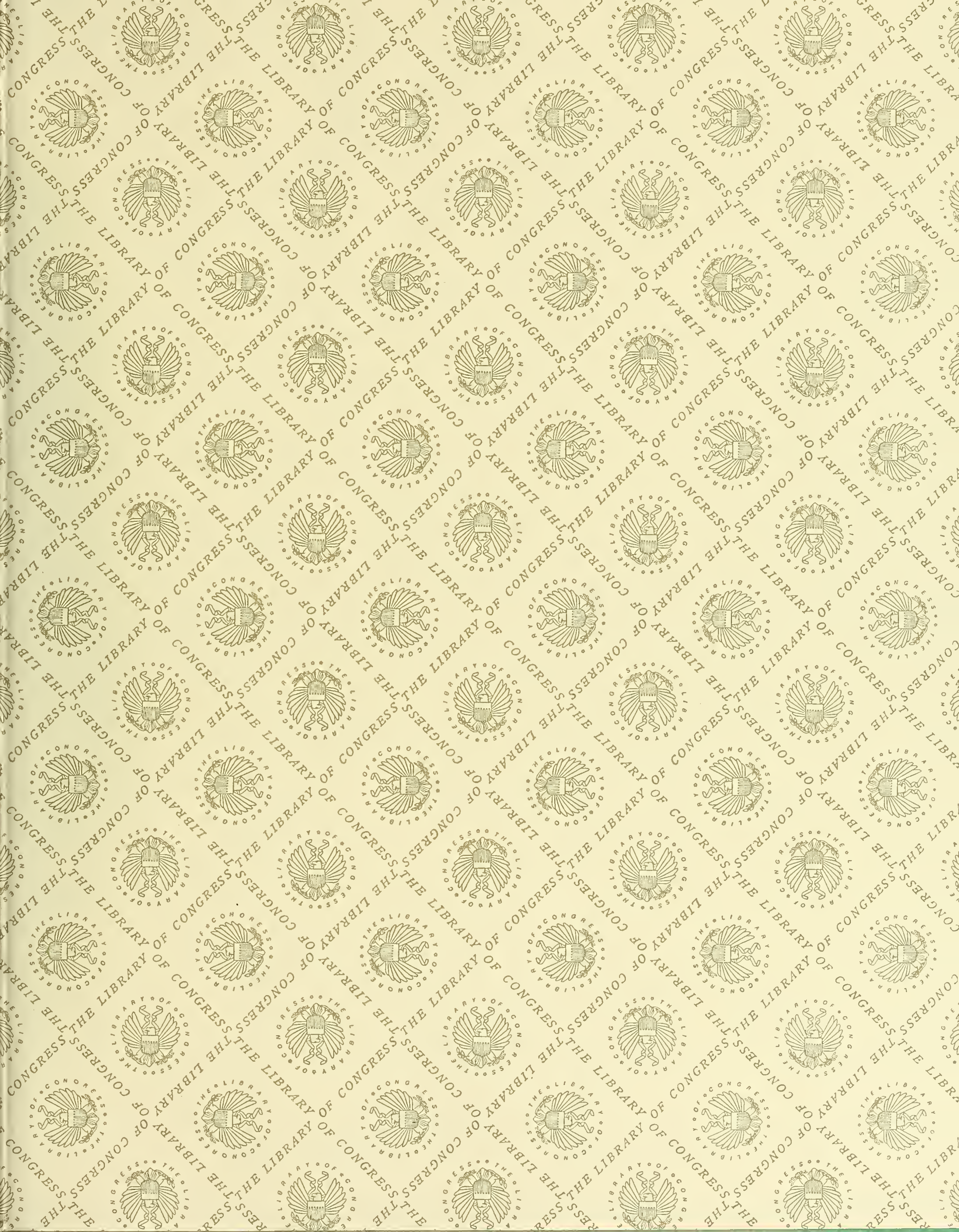

THE LIBRARY OF CONGRESS

IC 9062

Bureau of Mines Information Circular/1986

Microcomputer-Based Instrumentation System for Monitoring Ground Support in a Deep Mine Shaft

By J. K. Whyatt and E. L. Hardin

UNITED STATES DEPARTMENT OF THE INTERIOR

(United States. Bureau of Mines)

Information Circular 9062

Microcomputer-Based Instrumentation System for Monitoring Ground Support in a Deep Mine Shaft

By J. K. Whyatt and E. L. Hardin

UNITED STATES DEPARTMENT OF THE INTERIOR
Donald Paul Hodel, Secretary

BUREAU OF MINES
Robert C. Horton, Director

Library of Congress Cataloging in Publication Data:

Whyatt, J. K. (Jeff K.)

Microcomputer-based instrumentation system for monitoring ground support in a deep mine shaft.

(Bureau of Mines information circular ; 9062)

Bibliography: p. 15-16.

Supt. of Docs. no.: I 28.27: 9062.

1. Ground control (Mining)—Measurement—Data processing. 2. Rock deformation—Idaho—Shoshone County—Data processing. 3. Silver mines and mining—Idaho—Shoshone County. 4. Microcomputers—Programming. I. Hardin, Ernest L. II. Title. III. Series: Information circular (United States. Bureau of Mines) ; 9062.

TN295.U4 [TN288] 622s [622'.25] 85-600257

CONTENTS

ILLUSTRATIONS

TABLES

UNIT OF MEASURE ABBREVIATIONS USED IN THIS REPORT

°F	degree Fahrenheit	Ω	ohm
ft	foot	%	percent
in	inch	psi	pound per square inch
kΩ	kilohm	s	second
lb	pound	V	volt
μF	microfarad	V dc	volt, direct current
min	minute	W	watt
mV	millivolt	yr	year

MICROCOMPUTER-BASED INSTRUMENTATION SYSTEM FOR MONITORING GROUND SUPPORT IN A DEEP MINE SHAFT

By J. K. Whyatt[1] and E. L. Hardin[2]

ABSTRACT

This report describes a microcomputer-based instrumentation system for structural monitoring of the circular, concrete-lined Silver Shaft in northern Idaho. The 1-ft-thick concrete lining and surrounding rock mass were instrumented with a total of 12 multiple-point borehole extensometers, 16 pressure cells, 22 concrete embedment strain gauges, and 12 thermistors. The instruments were distributed among four test sites and installed immediately above the shaft bottom during sinking at the 2,414, 4,063, 5,191, and 5,955 levels. Limited access to the instruments prompted the development of a microcomputer-based remote data acquisition system that could be operated from the surface. The system scanned the instruments on a programmed schedule, printed out reduced data, and stored the data for plotting and recall. Data acquisition from off-site and uploading of data to mainframe computers were accomplished by the addition of modems to the basic system. The system is suitable for use in monitoring the structural behavior of any deep mine opening, especially in harsh operating environments and where access is limited.

[1] Mining engineer, Spokane Research Center, Bureau of Mines, Spokane, WA.
[2] Geophysicist, Science Applications Inc., Las Vegas, NV.

INTRODUCTION

Bureau of Mines personnel instrumented Hecla's Silver Shaft near Mullan, ID, as part of a research program into shaft support design. A major task within this project was the development of a rugged instrumentation and data acquisition system for shaft conditions. Instruments were installed at the 2,414, 4,063, 5,191, and 5,955 levels in the shaft during sinking over a 3.5-yr period.

SHAFT CONDITIONS

Severe time and environmental constraints were major considerations for instrumentation system design. The Silver Shaft was on the critical path for developing new areas of Hecla's Lucky Friday Mine, so interference with sinking operations had to be held to an absolute minimum. Three days were allotted for installation at each level, one of which was always a holiday. No time was allowed for in-shaft troubleshooting or manual data collection. Shaft environmental conditions were quite hostile to installation and operation of the instrumentation system and included:

1. Limited access.
2. Cramped space.
3. Humidity, 100 pct.
4. Corrosive, highly conductive water inflow.
5. Rock temperature, 120° F.
6. Shock and flyrock from blasting.
7. Mud, dust, and falling debris.

OBJECTIVES

The objective of this shaft design research project was to monitor rock mass behavior and the loads induced in the concrete lining during and after sinking of the Silver Shaft. The data were used by Bureau researchers to determine design criteria and establish models that predict changes in shaft behavior with variations in stress field, rock properties, and support system. In addition, the data collected by the instrument system were furnished to the shaft sinking contractor and mining company for internal use.

Application of ground support design experience acquired at other shafts is difficult. Differences in rock properties, geologic structure, in situ stress field, and the proximity of surrounding excavations may result in significantly different results for similar support methods. The Bureau is presently working to bridge this gap by developing quantitative records of shaft performance and improved predictive models for designing shaft ground support systems.

In summary, the instrumentation goals were as follows:

1. Provide accurate, timely data on rock mass and concrete lining behavior for on-site correlation with daily events and observations.
2. Establish a data base that would be useful for establishing shaft design criteria.

The main body of this report covers the data acquisition system, instrument specification, instrument modifications, and the installation plan. Instrument calibration, a program user's guide, and software listings are included as appendixes. Separate reports cover data analysis (1-2).[3]

ACKNOWLEDGMENTS

J. R. McVey, supervisory electronics technician, and P. A. Edminster, electronics technician, of the Bureau's Spokane Research Center assisted with microcomputer and data acquisition hardware specification and equipment maintenance. R. Muhs, mining engineer, also of Spokane Research Center, was the lead on-site person for routine data collection activities, liaison with Hecla personnel, and

[3] Underlined numbers in parentheses refer to items in the list of references preceding the appendixes.

equipment operation and maintenance. Mr. Pete Boyko, project manager, Hecla Mining Co., provided all necessary arrangements for use of Hecla facilities and equipment, and for installation of instruments in the shaft.

INSTRUMENT SPECIFICATION

Measurements indicative of the behavior of the supported rock mass, the effectiveness of the ground support system, and the design strength safety factor of the ground support system were needed to meet project goals. Radial closure of the shaft was anticipated to far exceed any tangential displacement, and would provide valuable data for judging the stability of the shaft. Elasticity theory predicts a reduction of radial closure rate with continued shaft advance; however, if closure accelerated, or even failed to slow significantly, the concrete lining would be in danger of failing.

The effectiveness of a ground support system could be determined by measuring the increase in rock mass cohesion and the confinement provided at free faces. Cohesion of the rock mass is increased by installation of bolts that intersect planes of weakness and prevent block movement. The performance of bolts in this manner is hard to measure with available instrumentation. A second way the bolts provide support is by applying confining pressure to free faces. This pressure can be determined by measuring the tension in individual bolts with strain gauges or rock bolt load cells. The Silver Shaft utilized Split Set[4] and grouted rock bolts, which are essentially passive, since they require rock movement for developing tension. Unfortunately, the variability of bolt performance and the difficulty in providing survivable instruments for the bolts precluded their instrumentation.

The second part of the ground support system was the concrete lining installed throughout the shaft. The lining provided confinement to the rock mass and sealed the surface to prevent rock degradation and spalling. As with the bolts, the lining requires shaft closure for developing a support load; however, the lining was relatively brittle and could rupture if overloaded, a problem not encountered with the bolts. Sampling of the hoop (tangential) stress yielded an estimate of the support load and strength safety factor.

Thus, the primary measurements sought from the Silver Shaft were the radial closure of the shaft and the hoop stress in the concrete lining. Instrument specifications required ruggedness, dc voltage output compatible with the data acquisition system, sensitivity, and appropriate operating ranges.

CLOSURE

The simplest method of measuring closure is with a tape extensometer. This method, and any other requiring shaft access or installation of a wire or rod across the shaft, were unsuitable because of interference with sinking operations. Closure was measured indirectly with borehole extensometers, which used deep anchors as a stable reference. In addition to the deep reference and collar anchors, other anchors were installed to provide a radial deformation profile of each borehole. The profile was useful for delineating zones of block movement and rock yielding, which are important in designing shaft support, as well as for confirming the stability of the deepest anchor.

Positioning the deepest anchor and specifying the range and accuracy of transducers required a prediction of the radial displacement as a function of distance into the shaft wall and distance to the shaft bottom. This estimate was made for the first installation with the elastic solution (3) as follows:

[4] Reference to specific trade names and manufacturers is made for identification purposes only and does not imply endorsement by the Bureau of Mines.

$$U_r = \frac{1-\nu^2}{E}\left[\left(\frac{S_x+S_y}{2}\right)\left(r+\frac{a^2}{r}\right)+\left(\frac{S_x-S_y}{2}\right)\left(r-\frac{a^4}{r^3}+\frac{4a^2}{r}\right)\cos 2\theta\right]$$
$$-\frac{\nu(1+\nu)}{E}\left[\left(\frac{S_x+S_y}{2}\right)\left(r-\frac{a^2}{r}\right)-\left(\frac{S_x-S_y}{2}\right)\left(r-\frac{a^4}{r^3}\right)\cos 2\theta\right] \qquad (1)$$

where U_r = radial displacement

S_x = maximum horizontal stress

S_y = minimum horizontal stress

E = modulus of deformation

a = shaft radius (without lining)

r = radius of depth of interest

ν = Poisson's ratio

θ = angle from S_x direction.

Examination of equation 1 revealed that radial deformation would decay rapidly with increasing radius. Figure 1 is a plot of this behavior for typical 2,414-level rock properties and stress field. The two curves represent the total and measurable elastic deformations. Measurable deformation is the 10% to 30% of the total that occurred after extensometer installation (4). The change in displacement with radial distance (i.e., slope) approached zero at about 45 ft, or twice the unlined shaft diameter of 22 ft. Thus, for purely elastic behavior, an anchor at 2 diameters could be considered stable. To be conservative, the deep anchor was placed at 55 to 60 ft, while the intermediate anchors were placed at 2, 5, 10, 15, and 30 ft to provide the deformation profile.

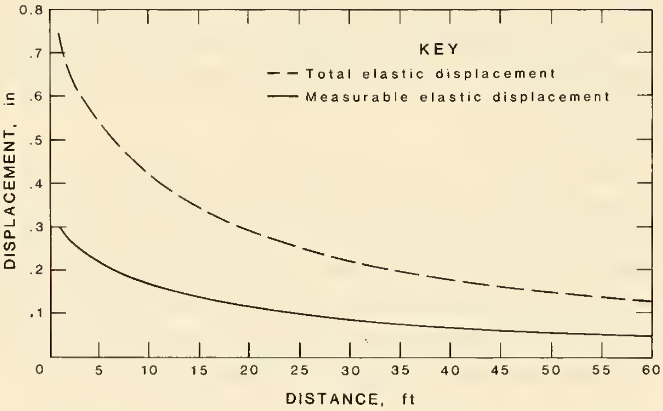


FIGURE 1. - Total and measurable rock displacement at the 2,414 level.

Extensometer accuracy was limited primarily by rod friction effects to about ±0.005 in (5), while data acquisition system resolution was limited to 0.1% of total range. In order to provide range for possible outward movement and extra closure, 2-in ranges were chosen for the first two levels. To allow for inelastic rock deformation, 4-in ranges were specified for the last two levels. A schematic of the extensometer is shown in figure 2.

LINING PRESSURE

Measurement of tangential lining pressure was complicated by the distribution of superimposed localized bending and shear stresses. Shear stresses were avoided by virtue of the flatjack pressure cell design chosen. The influence of bending stresses was minimized by positioning the cells to measure tangential stress at the middle of the lining. The tangential load produced by a given rock displacement was estimated by combining thin-wall cylinder solutions for both radial displacement and tangential stress due to an external hydrostatic pressure to give

$$S_t = U_r E/a \qquad (2)$$

where S_t = average tangential (hoop) stress

a = inside radius

E = modulus of deformation

U_r = radial displacement.

For a displacement of 0.005 in, the tangential stress in the liner would

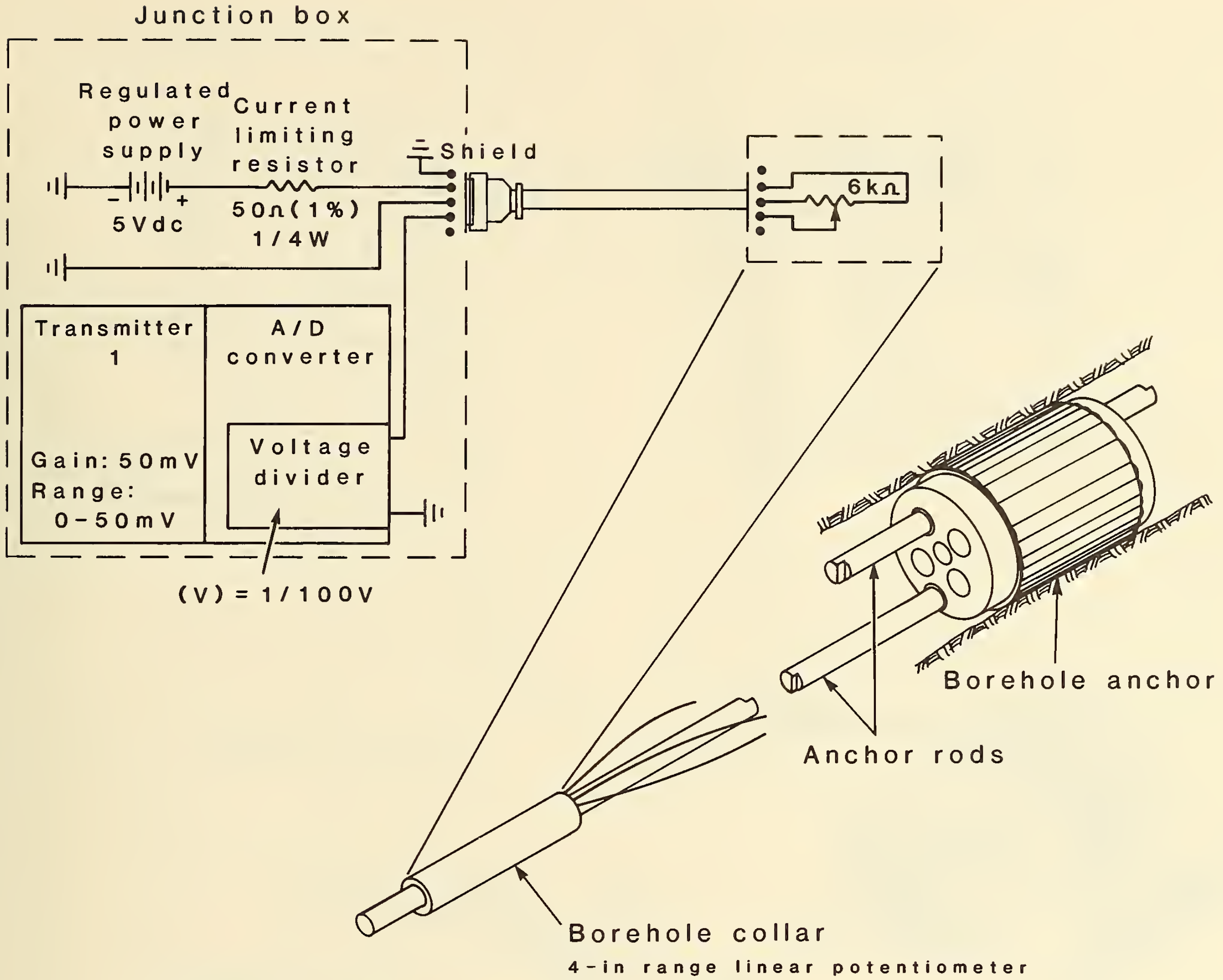


FIGURE 2. - Electrical schematic of the multiple-point borehole extensometer.

increase by 162 psi. Pressure cells of the type used have a sensitivity of within 10 to 15 psi, which is the limiting factor for ranges up to 10,000 psi, where the 0.1% resolution of the data acquisition system becomes important. The range of cells used corresponds to the expected stress range of 0 to 3,000 psi. Laboratory tests of concrete cylinders showed a uniaxial compressive strength of around 3,500 psi. A schematic of the pressure cell is shown in figure 3.

LINING STRAIN

Concrete typically displays an increasing strength and deformational modulus with time, suggesting that lining strain is load-path dependent. Early loading of the lining while still green may be relieved by creep of the concrete lining. Finding the amount of creep and the effect of increasing modulus requires measurement of strain in the concrete lining. The strain readings were also valuable as a backup in case of pressure cell failure. Concrete embedment strain gauge specifications are comparable with those of the pressure cells, with a sensitivity of ±20 microstrain and a range of 0 to 20,000 microstrain. Figure 4 is a schematic of the embedment strain gauge circuit.

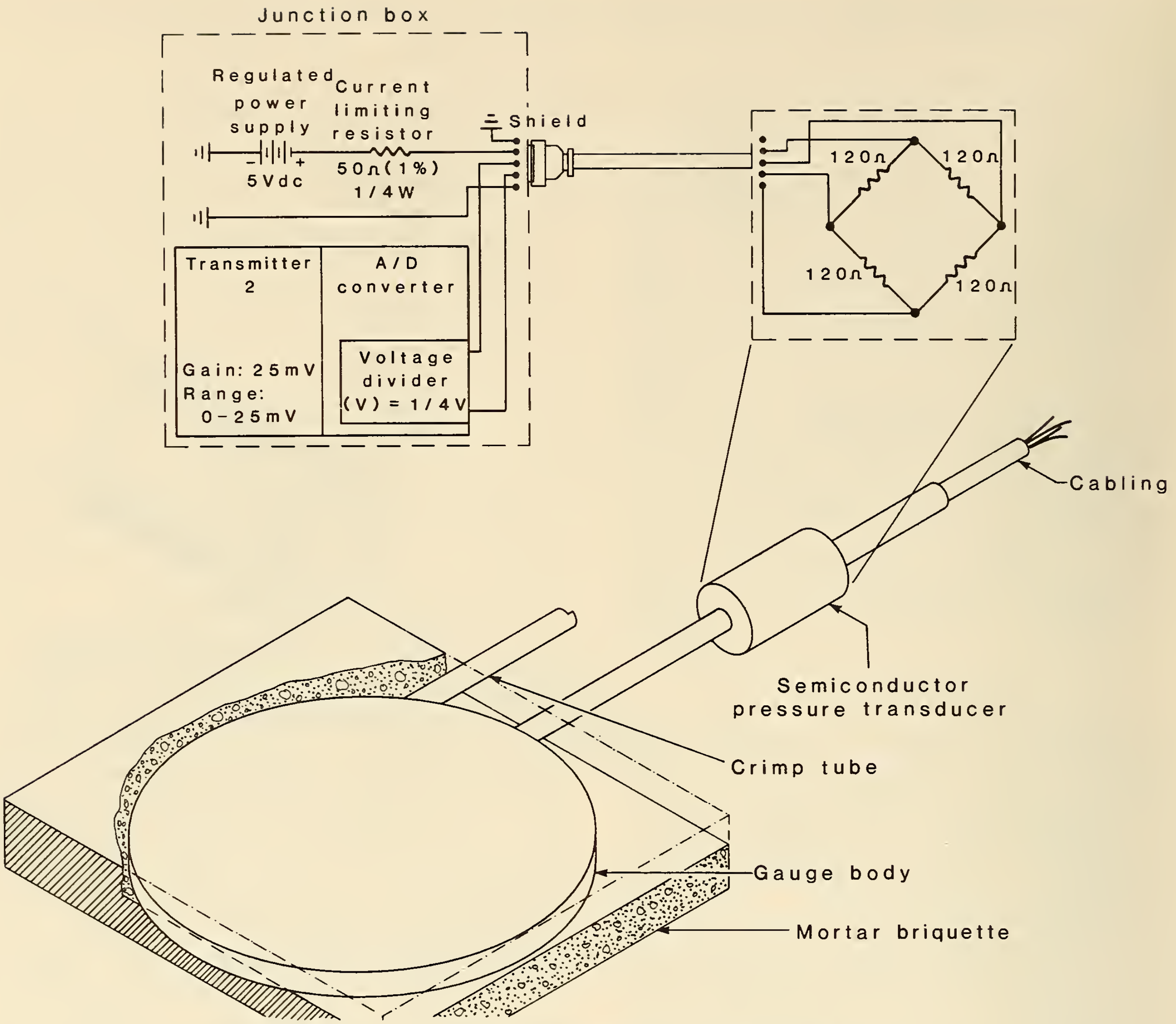


FIGURE 3. - Electrical schematic of the pressure cell.

TEMPERATURE

The grout curing in the extensometer boreholes and the concrete lining produced significant heat and elevated the temperature of the instruments for a short time after installation. Although the instruments were expected to function at the expected maximum temperatures, temperature was monitored for correction of instrumentation data. Thermistors were specified to handle a range of temperatures from 60° to 160° F and for durability. The schematic for the thermistor is shown in figure 5.

INSTRUMENT MODIFICATIONS

The instruments installed in the Silver Shaft were mostly of proven designs that were available "off the shelf," although they required modification for efficient installation and protection against shaft conditions. These modifications evolved with the project, resulting in the designs presented here.

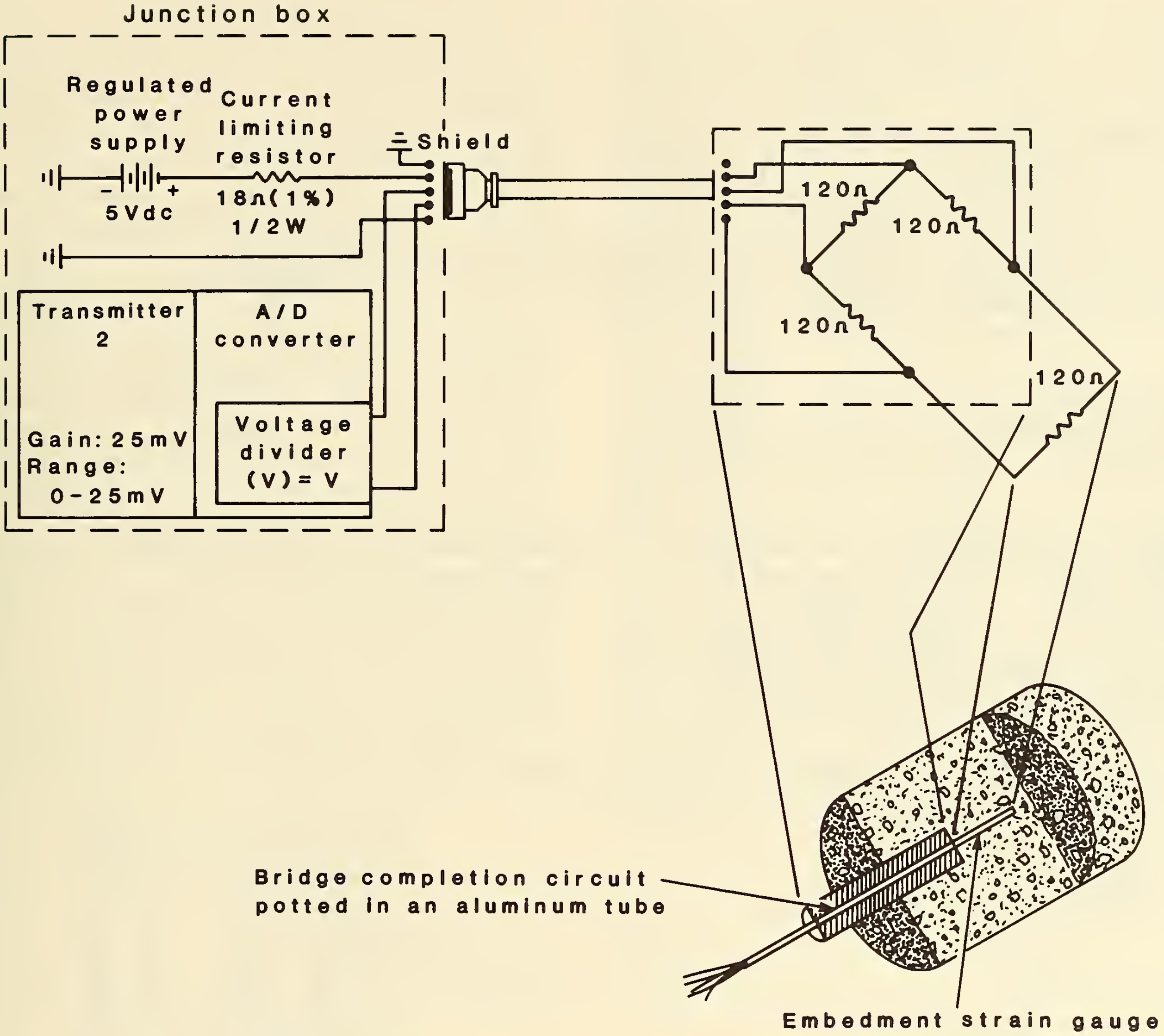


FIGURE 4. - Electrical schematic of the concrete embedment strain gauge.

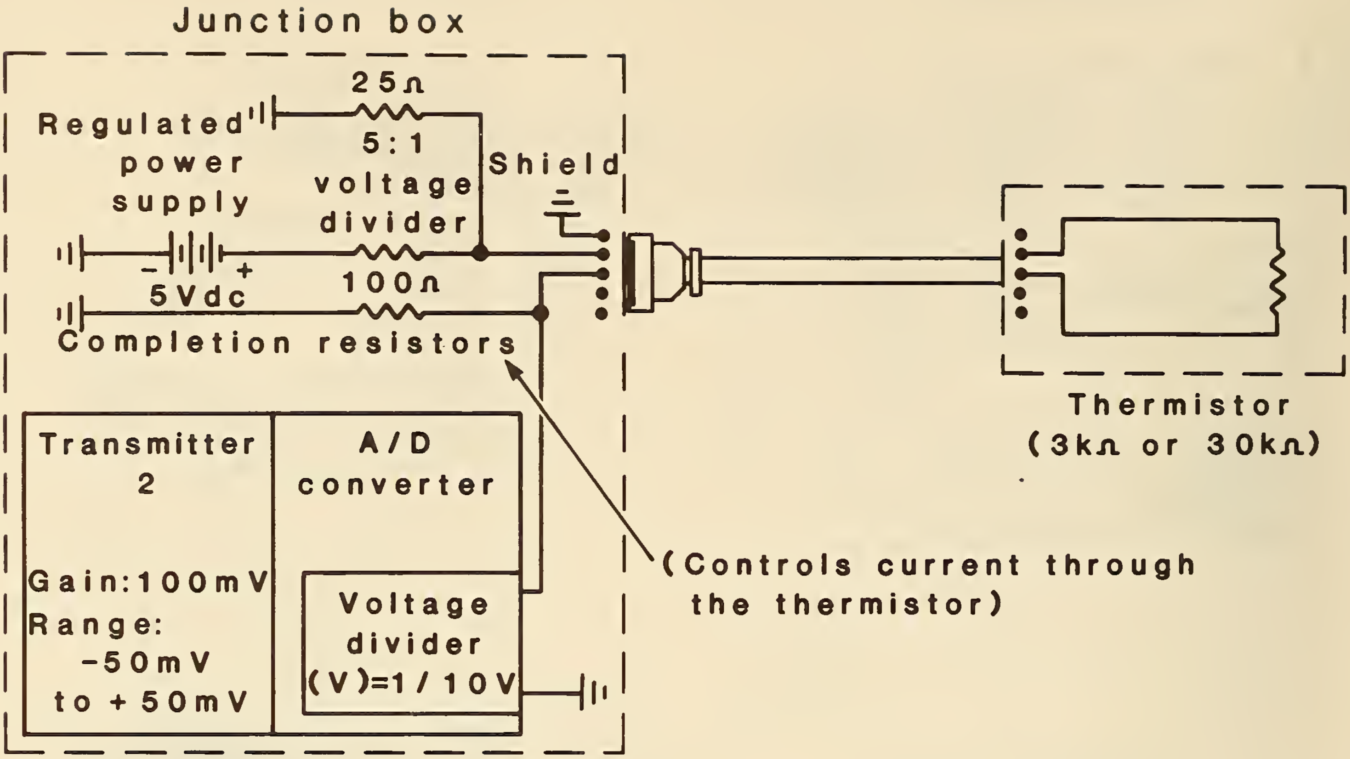


FIGURE 5. - Electrical schematic of the thermistor.

EXTENSOMETERS

Geokon model A3 multiple-point borehole extensometers with inflatable anchors were modified for ease of installation and increased protection against lining installation, blasting shock, and flyrock. These modifications did not affect the operation of the instrument but were additions to the packaging of system components.

The limited time and space available at the shaft bottom demanded that the extensometers be preassembled at the surface, lowered to the bottom of the shaft beneath the muck bucket, and inserted into the borehole while still hanging beneath the muck bucket (fig. 6). The sling for suspending the extensometer protected the displacement transducers in the head from loading by the weight of the anchors and anchor rods, while it preserved the flexibility of the instrument for bending into the borehole. The sling consisted of a steel cable attached to the head of the extensometer to support the blast protection enclosure and electronics, and a polypropylene rope that was run along the length of the instrument, tied off at each anchor and at the head, in order to support the weight of the rods and anchors. A 50-lb weight tied to the bottom of this rope minimized swinging while the instrument was lowered to the bottom of the shaft. The assembly was lowered at slow speed and monitored for fouling in the shaft steel. The sling was easily disassembled during instrument installation.

Protection against blasting shock and flyrock damage to the extensometer and readout cables was provided as shown in figure 7. The major components were the head enclosure can, the blast shield, and the cable armor. The head enclosure was basically a steel can mounted to the base of the extensometer head to protect the transducers and electrical connections. Holes were provided for water drainage and exit of the signal cables. The enclosure also served to retain the grout in the borehole. The blast shield was a 3/8-in steel plate rockbolted over the extensometer head, which was recessed

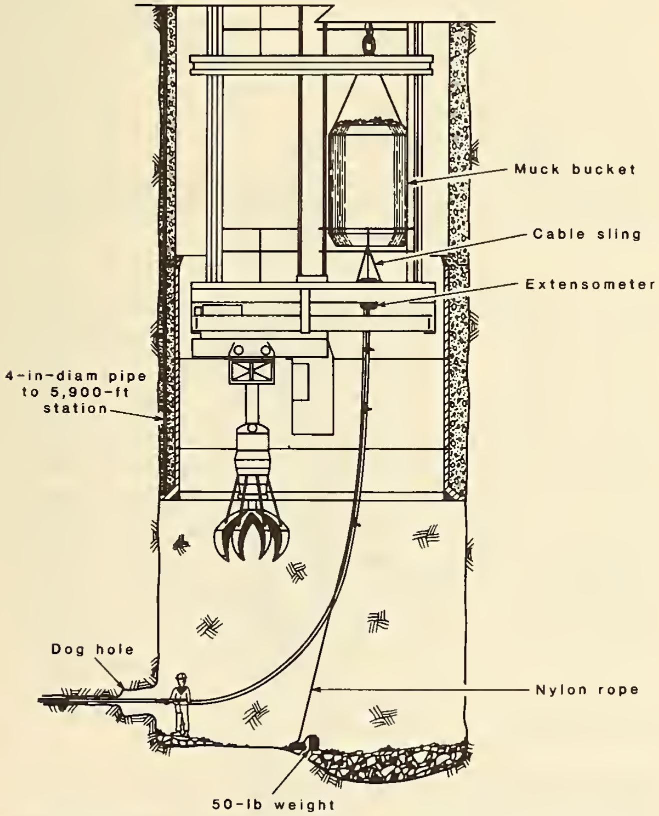


FIGURE 6. - Multiple-point borehole extensometer installation.

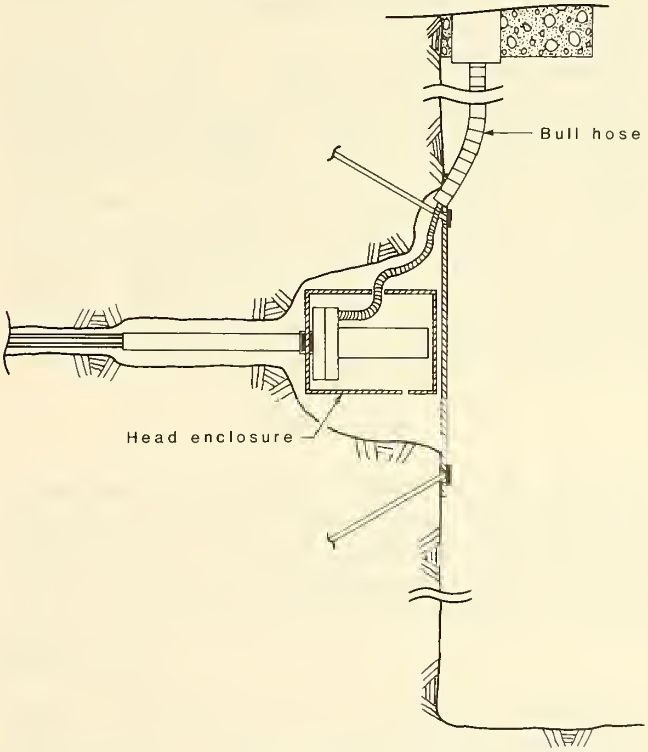


FIGURE 7. - Multiple-point borehole extensometer head protection.

into a "doghole." The signal cables were delivered in 3/8-in conduit with waterproof connectors. Additional cable protection was provided by the head enclosure and blast shield at the extensometer head, and a covering of 3-in compressed air bullhose tied to the wire mesh, which extended up to the concrete lining. The cables were then run through a 4-in schedule 40 pipe in the lining to the junction box on the station above.

PRESSURE CELLS

Geokon model 3600, 9-in-diam, mercury-filled pressure cells with semiconductor pressure transducers were precast in grout blocks for precrimping in the lab. The crimp tube of the pressure cell is used to expand the cell and establish positive contact with the concrete. Contact is often lost during curing of the concrete because of water migration to the steel and differential thermal expansion. Precasting enabled contact to be established in the laboratory instead of the shaft, and eliminated the need to expose the crimp tube to shaft conditions. The stiffness contrast between the grout and cell required a final calibration (appendix A).

Anchor wires were cast along with the cells into 13- by 13- by 4-in blocks, which tended to float in the liner concrete. The wires were tied to 18-in, U-shaped rebar anchors placed in the concrete. The cell cables were encased in 3/8-in flexible conduit that ran up the shaft wall to a 4-in pipe in the liner pour above, and finally to the junction box on the station above.

STRAIN GAUGES

Four-inch Ailtech Concrete Embedment Strain Gauges were precast in a number of different briquettes for protection against damage during installation. The final briquette design was a 3-in-diam by 6-in cylinder cast using the shaft concrete mix (large aggregate removed) 3 days before installation (fig. 4). Tie wires were cast through the briquettes for anchorage during installation. The anchors were similar to the pressure cell anchors. Bridge completion circuits were built in 1/2-in aluminum conduit, potted in place, and cast in the ends of the briquettes. The strain gauge cables were protected by flexible conduit and wired in groups of five, as shown in figure 8.

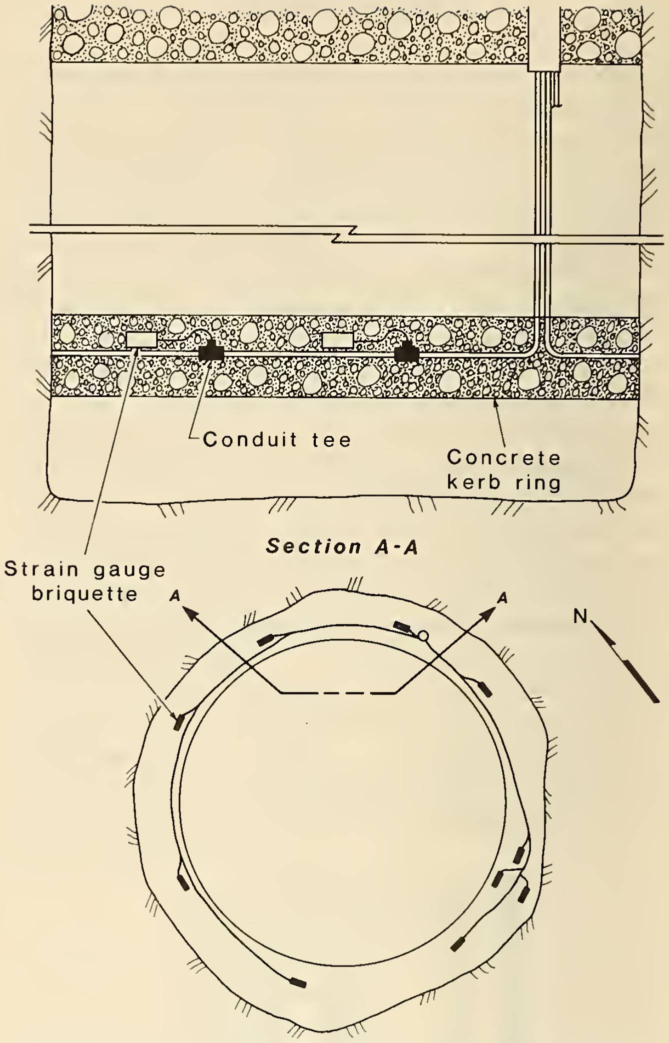


FIGURE 8. - Encapsulated strain gauge installation, 5,960 level.

DATA ACQUISITION SYSTEM

A primary concern in designing the instrumentation system for the shaft was ensuring accurate, timely collection of data from the instruments. Previous experience (6-7) with shaft data collection and anticipated shaft conditions indicated the need for a remote data acquisition system (RDAS). The goals of the Silver Shaft project necessitated frequent readings for several months' time.

The operating specifications of the RDAS included frequent scanning of the instruments, preservation of instrument accuracy, and transmission of data to the surface over distances up to 6,800 ft. In addition, the system had to provide instrument data quickly and clearly to mine engineers and had to be easy to use. Finally, the shaft cable requirements were minimized to control costs and installation time, especially for the deeper levels.

The RDAS was based on a Bureau of Mines design (7) for a blind shaft boring machine, which included an underground subsystem to read instruments and transmit data to the surface, and a surface subsystem to receive, reduce, and store the data. The underground subsystem included an instrumentation power supply, voltmeters, analog to digital (A/D) converters, and digital transmitters enclosed in a sealed junction box. The surface subsystem included a digital receiver, data buffer, and microcomputer. The system and instrumentation levels are illustrated in figure 9.

UNDERGROUND SUBSYSTEM

The underground subsystem for each level was built around two Burr-Brown Micromux #6820 transmitters. Each transmitter measures voltage outputs from 16 instruments and transmits the readings to a receiver on the surface over a single twisted-pair telephone cable. The transmitters operate automatically, continuously scanning instruments and transmitting data.

The transmitters, and a 5-V dc regulated power supply for instrument excitation, were sealed in a 24-in (wide) by 36-in (high) by 12-in (deep) stainless steel junction box (J-box) for protection against corrosion, shock, and invasion. Instrumentation, mine power, and transmission line connections into the box were made through waterproof bulkhead connectors. A 100-W heater inside the box was operated continuously to maintain

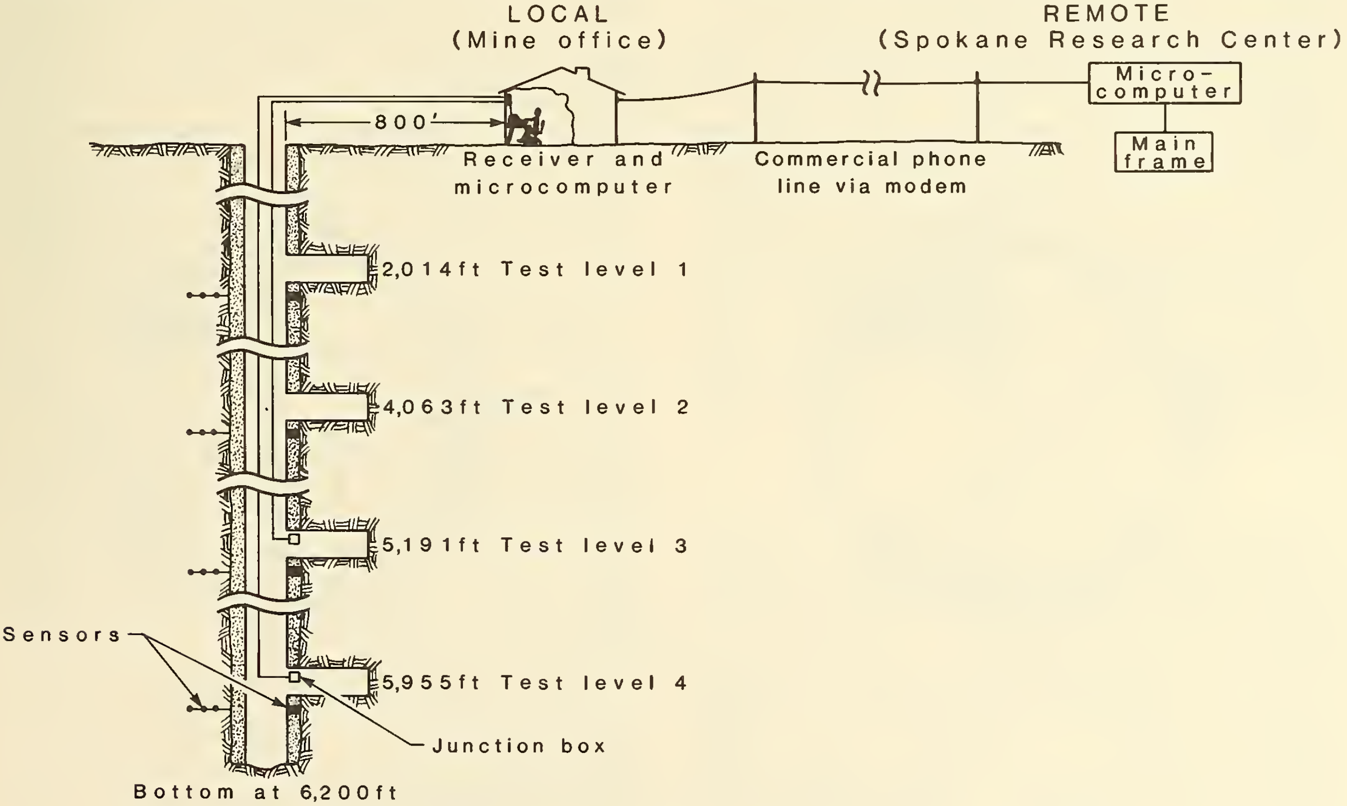


FIGURE 9. - Data acquisition system configuration at the Silver Shaft.

a slightly elevated temperature and prevent condensation. Each J-box was installed in a nearby station, as close as possible to the instrumented level.

SURFACE SUBSYSTEM

The twisted-pair telephone cables from the transmitters were run up the shaft, through a buried PVC conduit to the mine office, and into a Micromux #6800 receiver. The receiver collected the transmitted data as ASCII code in a buffer that was refreshed every 10 s (for a two-transmitter system). The contents of the buffer were available for reduction and storage by a microcomputer.

A software package, MDAP85, was developed for the Hewlett-Packard HP-85 microcomputer to collect and process the buffered data. Specifications of the package include:

o Automatic data sampling time intervals from 3 min to several hours (all channels).

o Immediate data reduction and printout.

o Produce on-site CRT or hard-copy plots of results.

o 100-scan automatic storage capacity.

o Long-distance automatic data collection through modems.

o Data file uploading to mainframe computers via modems.

o User friendly.

o Capable of monitoring two instrumentation levels.

A detailed user's guide and program listings are included as appendixes B and C, respectively.

INSTALLATION PLAN

The instrumentation levels were located so as to provide a profile of changing behavior with depth while focusing on deeper, and hence more highly stressed, sections of the shaft. The geology was fairly consistent with depth, except for occasional minor faults and joint sets that were not mapped beforehand, and played little part in locating the instrument levels. However, the orientations of some of the extensometer boreholes had to be slightly adjusted where weak, ravelly rock or faulting was encountered. Since the instruments had to be installed with a minimum of interference with sinking operations, the exact depth of each level was determined by the timing of holidays when sinking was halted.

2,414 LEVEL

The first set of instruments, consisting of multi-point borehole extensometers, concrete pressure cells, and thermistors, was installed at a depth of 2,414 ft. Definition of radial deformation as a function of angle around the shaft was sought to determine the influences of bedding, rock properties, and in situ stress field. This suggests the installation of six extensometers at 60° angles, which was considered too expensive and time consuming to be practical. Instead, three extensometers were specified, as this would provide sufficient data to determine the effects of major discontinuities and the in situ stress field. The total closure measurements depended on the symmetry of the shaft, although the closure at each extensometer towards the center of the shaft was estimated from the stable anchor. In order to study the influence of bedding on deformation, the extensometers were oriented parallel, perpendicular, and at 45° to the bedding strike.

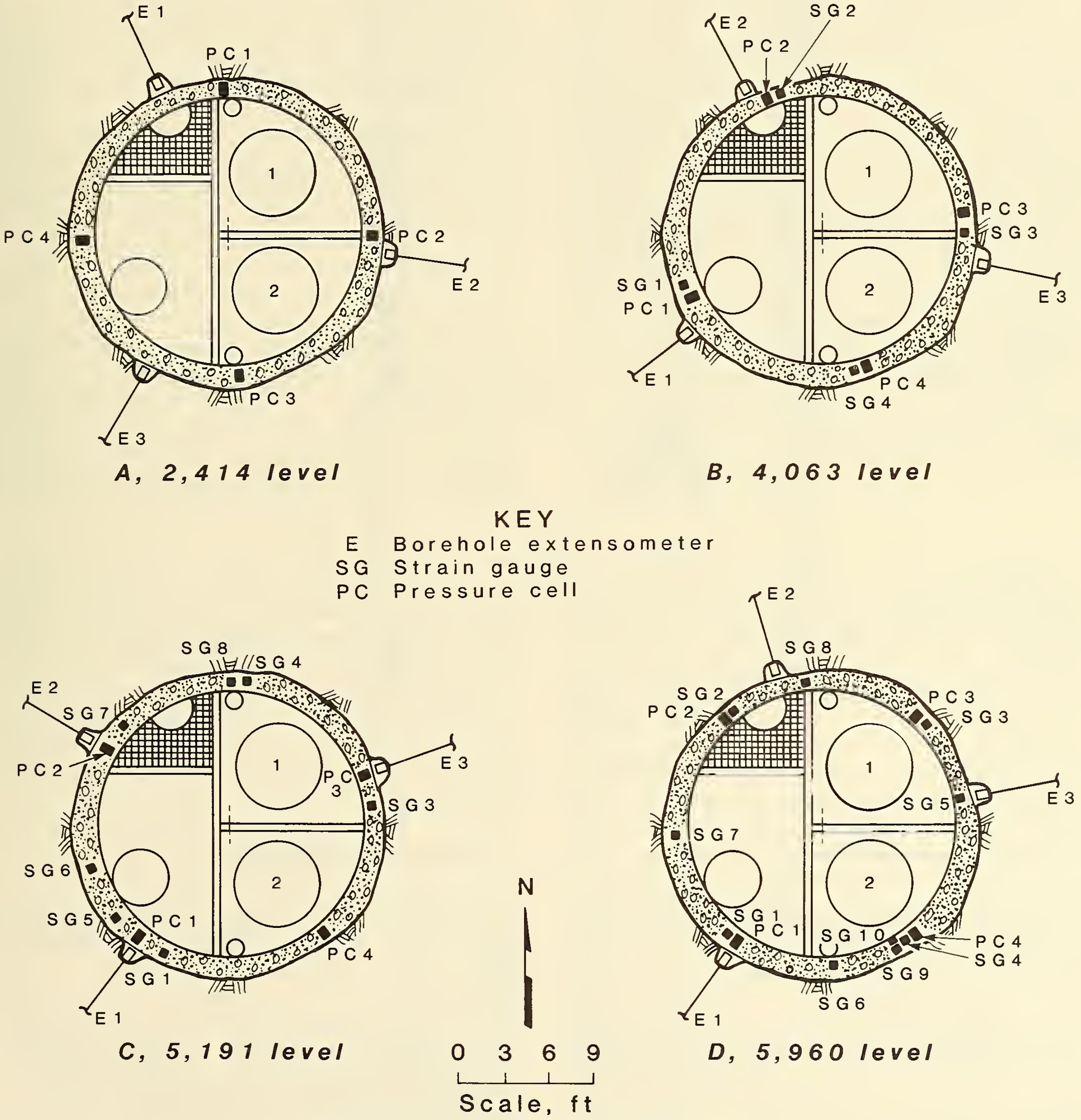
The pressure cells monitored hoop stress in the lining. In an ideal lining with no shear on the rock-concrete interface, opposite sides of the lining carry identical loads and the magnitude of the loads is directly related to the in situ stress field. In order to directly compare measured liner stress with the predictions of idealized models, four pressure cells were installed at 90° angles starting with shaft north.

Temperature variations due to curing of borehole grout and the lining concrete were measured with thermistors at each of the pressure cells, at one of the extensometer heads, and at three points along one of the extensometer boreholes.

The extensometers were installed as close as possible to the shaft bottom (about 4 ft). The kerb ring form was then lowered to the shaft bottom, and the pressure cells were installed with the concrete pour. Installing the 1-ft-thick concrete lining to within 4 ft instead of the usual 20 ft, of the shaft bottom increased the maximum possible elastic load imparted to the lining. Plan and section views of the instrument configuration are shown in figures 10*A* and 11, respectively.

4,063 LEVEL

The instrumentation plan was modified at the 4,063 level by eliminating the



A, 2,414 level

B, 4,063 level

C, 5,191 level

D, 5,960 level

FIGURE 10. - Instrument location by level.

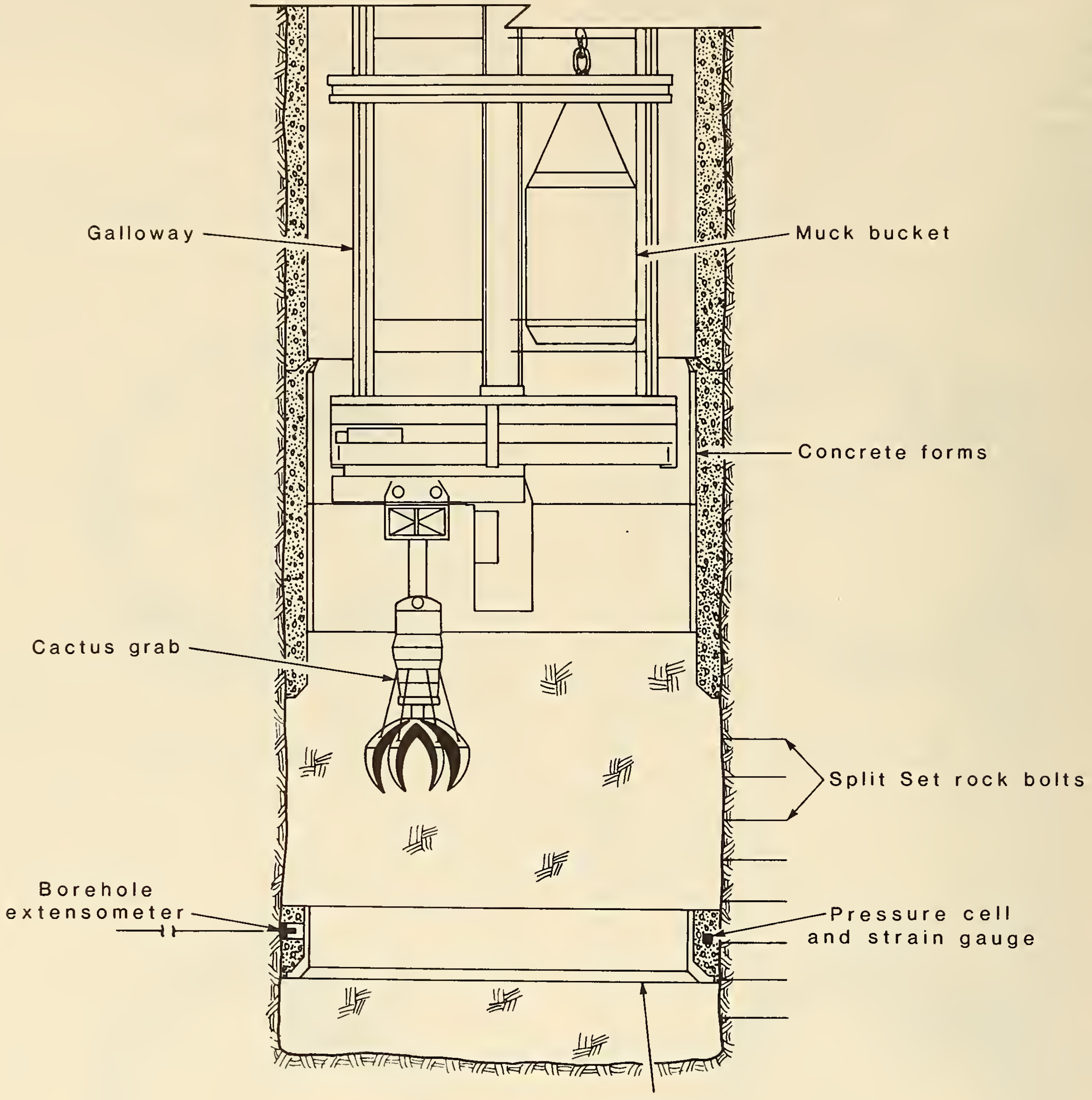


FIGURE 11. - Instrument installation at the 2,414 and 4,063 levels.

borehole thermistors, which showed little response at the 2,414 level, and adding concrete embedment strain gauges (fig. 10*B*). These gauges were installed next to each pressure cell to measure the strain equivalent to the stress measured by the pressure cells. Creep strains could then be estimated by assuming a value for liner modulus and substracting load cell strain estimates from strain gauge readings. In order to avoid damaging the gauges during installation, they were precast in 3- by 5- by 6-in briquettes of fine aggregate concrete and 98% cured before installation. The addition of the briquette complicated data interpretation by introducing a constantly changing modulus contrast between the curing concrete and the more fully cured briquette.

5,191 LEVEL

Changes were made in instrument selection and placement, as well as installation procedure, for the third level. In anticipation of plastic rock behavior, liner installation was delayed 2 days for 20 ft of additional shaft sinking, which is the normal procedure. The amount of elastic deformation "felt" by the lining was thereby reduced, but the sequence was more representative of the normal sinking cycle. In addition, the range of the extensometer transducers was increased to 4 in.

The selection of instruments was modified by replacing the four pressure cell thermistors with unencapsulated strain gauges (fig. 10*C*). Data from the first two levels had well defined the temperature changes due to the curing concrete, and additional data were needed on concrete lining performance. The unencapsulated gauges were installed in an attempt to judge the effects of the precast briquettes. Ceramic aggregate was used to reduce the strain gauge briquette modulus in order to increase gauge sensitivity and more nearly match lining modulus during the first few days after installation.

5,955 LEVEL

Further adjustments to instrument selection and placement were made for the final level (fig. 10*D*), but the installation procedure remained the same as for the 5,191 level. The extensometers were modified by removing the 2-ft anchors and converting to hydraulic anchors, but still grouting the borehole. The unencapsulated strain gauges were eliminated because of their excessive failure rate (75%), but the number of encapsulated gauges was increased from 4 to 10. Eight of the gauges were distributed at equal intervals around the shaft, and the remaining two were placed to the inside and outside of the liner to measure the strain gradient across the thickness of the lining. The briquettes for this level were 3-in-diam cylinders 6 in long, of shaft mix concrete that was poured 3 days before installation to minimize modulus contrast at any given time and to ensure that the final moduli were equal.

SUMMARY

An effective microcomputer-based instrumentation system for monitoring ground support in a deep mine shaft was developed and implemented. Experience from four instrumentation levels was used to refine the instrumentation system to its present state. The system has operated without maintenance for months under harsh mine conditions, alleviating the need to send personnel underground, and providing results quickly. It is suited to most underground instrumentation projects, although modifications may be needed if transmission distance is substantially greater than 6,800 ft. Since the development of this system, it has become a standard at the Bureau's Spokane Research Center and is being used, with some modifications, for monitoring construction of a rectangular shaft and coal mine ground support. Implementation of the system has reduced personnel requirements in the field, increased the frequency and accuracy of data collection, and reduced mobilization time for field projects. Uploading of data to the Bureau's mainframe computer in Spokane has facilitated handling of the typically extensive data bases that result from these projects.

REFERENCES

1. Board, M. P., and M. J. Beus. Instrumentation and Preliminary Analysis of a 6,200-ft Deep Circular Shaft in Northern Idaho. Paper in Proceedings of the Symposium on Geomechanics Applications in Underground Hardrock Mining (Denver, CO, Oct. 20-26, 1984). Soc. Min. Eng. AIME, 1984, pp. 127-140.

2. Beus, M. J., and M. P. Board. Field Measurement of Rock Displacement

and Support Pressure at 5,955-ft Level During Sinking of Deep Circular Shaft in Northern Idaho. BuMines RI 8909, 1984, 11 pp.

3. Obert, L., and W. I. Duvall. Rock Mechanics and the Design of Structures in Rock. Wiley, 1967, p. 108.

4. Schwartz, C. W., and H. H. Einstein. Improved Design of Tunnel Supports: Volume 1 - Simplified Analysis for Ground-Structure Interaction in Tunneling. U.S. Dep. Transportation, Rep. DOT-TSC-UMTA-80-27.I, June 1980, v. 1, 450 pp.

5. Cording, E. J., A. J. Hendron, W. H. Hansmire, J. W. Mahar, H. A. MacPherson, R. A. Jones, and T. D. O'Rourke. Methods for Geotechnical Observations and Instrumentation in Tunneling (grant GI-33644x, Nat. Sci. Foundation). Univ. IL, Dep. Civil Eng., Champaign-Urbana, IL, 1975, 2 volumes, v. 1, 296 pp.

6. Gooch, A. E., and J. P. Conway. Field Measurement and Corresponding Finite-Element Analysis of Closure During Shaft Sinking at the Lucky Friday Mine. BuMines RI 8193, 1976, 19 pp.

7. McVey, J. R., W. C. Vilwock, and P. F. Sands. Computer Monitors Twenty-Four Foot Blind Shaft Boring Machine. Paper in Proceedings of the Western Mining Industry Electro-Technology Conference (Reno, NV, Sept. 14-16, 1981). Univ. NV - Reno, 1981, 24 pp.

APPENDIX A.--INSTRUMENT CALIBRATION

Although the instruments installed in the shaft were acquired commercially, some laboratory work was necessary to confirm factory calibrations, compensate for instrument modifications, and investigate possible sources of error.

LINEAR POTENTIOMETERS

All of the linear potentiometers were calibrated prior to the final assembly of the extensometers. The potentiometer was wired into the data acquisition system, clamped in a calibration jig, and calibrated with a micrometer. The raw data output was plotted against displacement in a least squares fit.

CONCRETE PRESSURE CELLS

The placement of concrete pressure cells in concrete often results in the formation of a gap on the steel-concrete interface. Migration of water from the hot, curing concrete mass to the cooler steel surface of the pressure cell may contribute to the problem. Another possibility is decoupling during cooling because of the greater thermal expansivity of the steel.

To overcome this problem, the pressure cells were first cast in a grout "brick." A cast pressure cell is shown in figure A-1. The grout mix design (table A-1) was developed specifically to match the fully cured modulus of the grout with that of the Silver Shaft concrete. Thus, the pressure cell was thought not to be a "high modulus inclusion" in the cured liner, which could concentrate stress.

With the grout brick in contact with the concrete in the liner, void production was minimized. A "cold joint" formed between the concrete and brick, but it was perpendicular to the applied compressive stress and probably did not influence the load transfer.

Each cell was cast in a grout brick with dimensions of 13-in by 13-in by 4-in thick. Following 4 h of curing, the cells were placed in moist polyethylene bags to allow a slow cure. After 7 days, the cell-grout assemblies were calibrated in a compression testing machine. The applied stress was brought up in 100-psi units, and the output voltage of the pressure transducer was recorded. An example of the voltage output versus stress is given in figure A-2*A*. When the cells exhibited nonlinearity at low applied stress (fig. A-2*B*), the pinch tube was crimped, forcing mercury into the cell.

TABLE A-1. - Pressure cell grout mix

	Parts by weight
Component:	
Cement (portland 1A)..	8
Water................	4
Sand.................	32

Two other calibrations were performed. First, the temperature sensitivity of the cell-grout system was examined. The assembly was placed in an oven at room temperature. The temperature of the oven was raised in approximate 10° F intervals, each time allowing the gauge output to stabilize. The results of this calibration are shown in figure A-2*C*. The purpose of this test was to examine any voltage offset due to temperature expansion of the various system components under zero applied stress. As seen in figure A-2*C*, a change in gauge output of only 0.35 mV was experienced. This change is 1% of total gauge output from 0 to 1,000 psi and represents a calculated stress change of only 10 psi. This negligible change indicates low thermal sensitivity of the instrument.

A second test was performed to examine any slope change in the calibration curves as a function of temperature. The cell was heated to 120° F, then removed from the oven while insulated; a calibration load cycle was then performed as quickly as possible. The subsequent calibrations indicated no significant change in the slope of the stress-voltage curve, so the stress-voltage curves at ambient temperature were used throughout the project.

THERMISTORS AND STRAIN GAUGES

The thermistors were calibrated by the manufacturer prior to shipment. Strain gauge reduction data were supplied by the manufacturer.

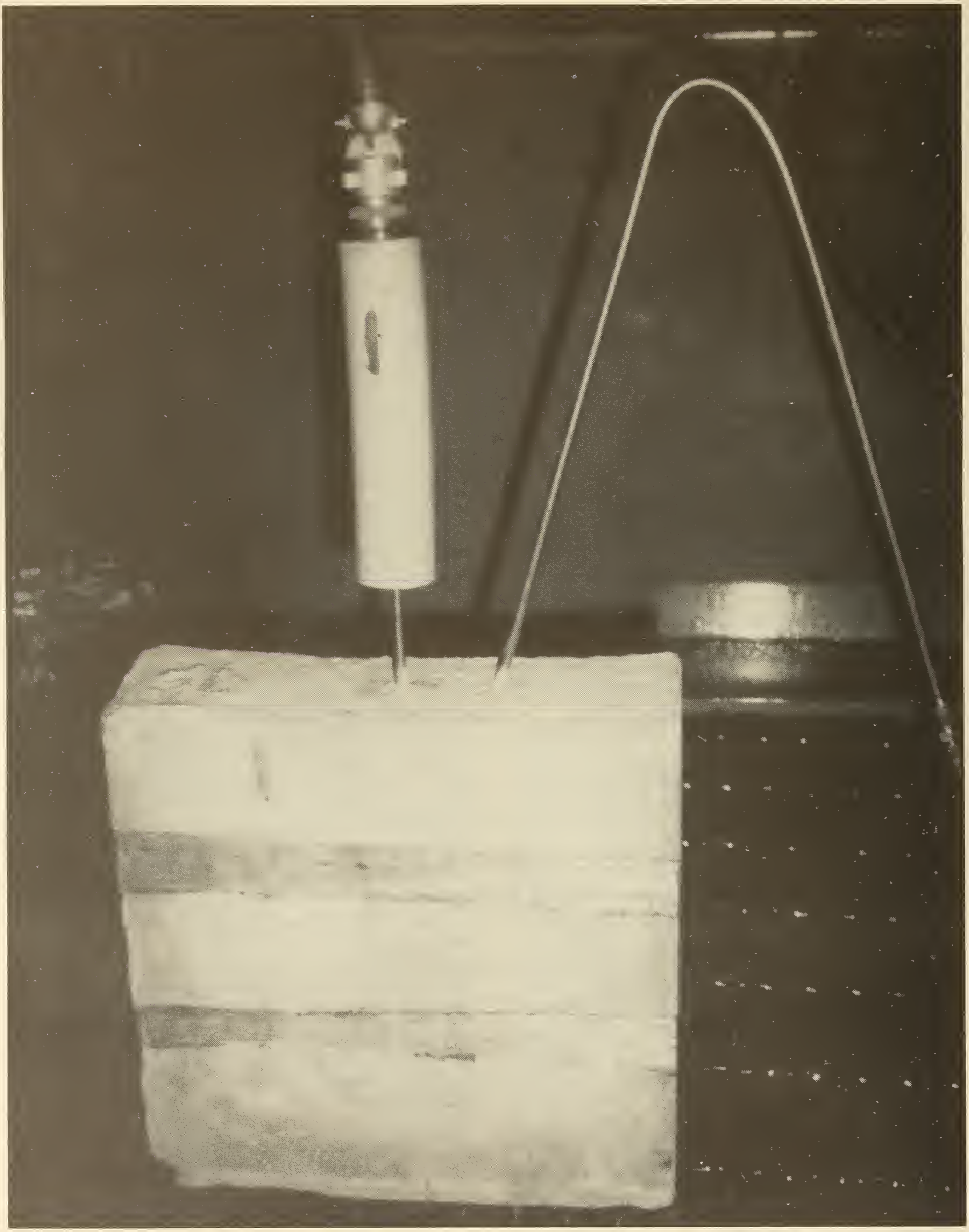


FIGURE A-1. - Cast pressure cell.

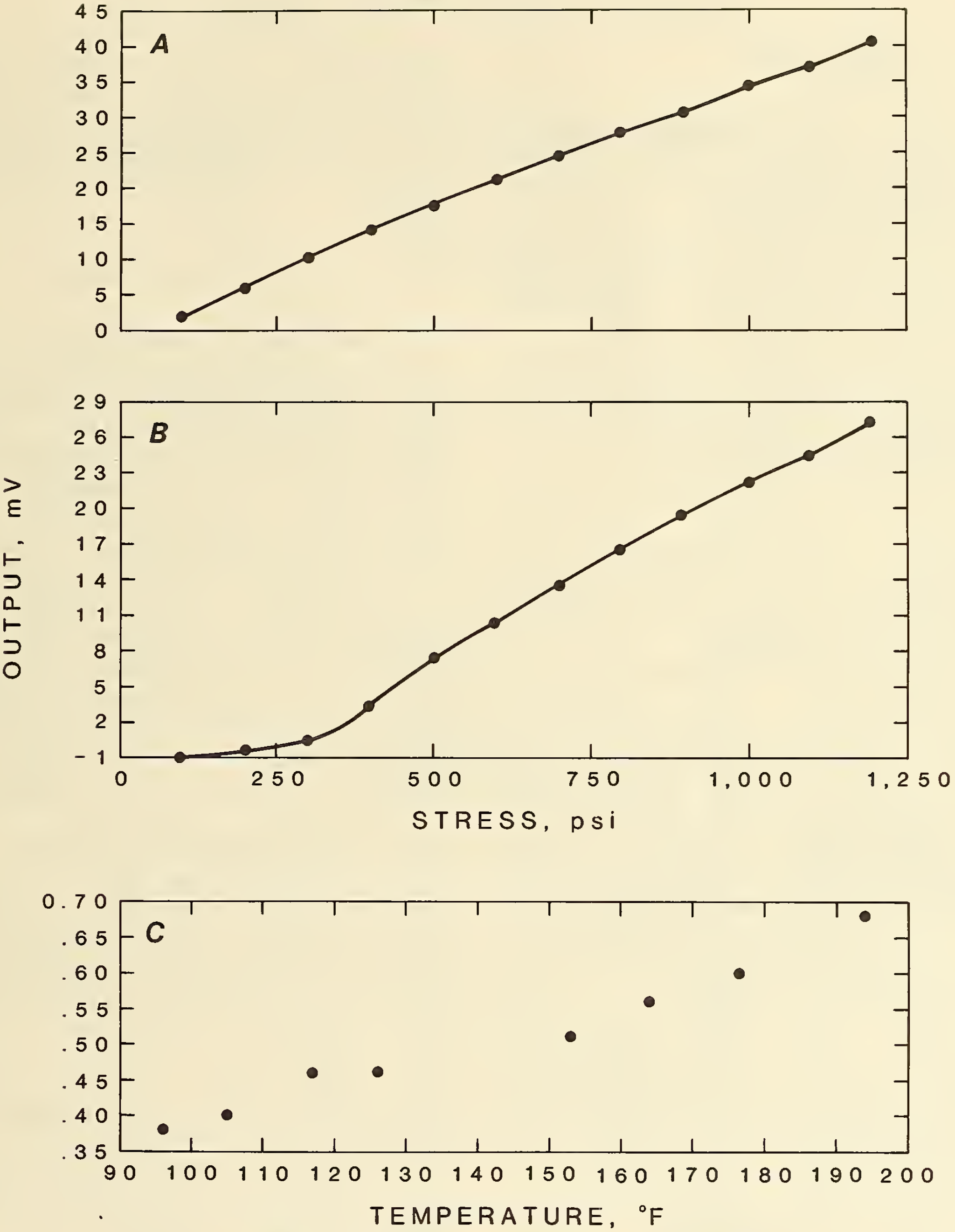


FIGURE A-2. - Pressure cell calibration curves.

APPENDIX B.--USER'S GUIDE TO MDAP85

INTRODUCTION

MDAP85 is a BASIC language program written for collecting and reducing data from a Micromux data acquisition system with an HP-85 microcomputer. The original version of this program was developed by Terra Tek Engineering under Bureau contract J0205048 entitled, "Structural Evaluation of a Circular, Concrete-Lined Shaft in a Deep Vein Mine." This program was enhanced by the Bureau for monitoring instrumentation at the 5,955 level of Hecla's Silver Shaft Mullan, ID.

Program enhancements made by the Bureau include:

1. Automatic data acquisition via telephone modem.
2. Uploading from HP-85 data tapes to a Burroughs 6800 mainframe.
3. Enhanced output including change in data since last scan.

MDAP85 can collect data from up to 32 channels from each of two different instrumentation sites for a total of 64 channels. MDAP85 consists of several different program modules, most of which are stored on a tape cartridge referred to throughout this manual as the "program tape." Other routines for long-term plotting and data uploading are stored on a tape cartridge referred to as the "utility tape." BASIC listings of all program modules are given in appendix C. The use and purpose of each program module, as well as the hardware and software required to utilize this program, are described in this appendix.

HARDWARE REQUIREMENTS

The hardware (figure B-1 and table B-1) is grouped into three subsystems: the Micromux transmitters underground, a local data acquisition system at the mine office, and a remote data acquisition system at Spokane Research Center. Additional remote systems could be added, although only one could collect data at a time. The arbiter (fig. B-2) was built to allow remote access on demand, but it momentarily prevents the local system from collecting data. The local and remote HP-85's are not aware of each other, so the mine office HP-85 can be eliminated if only remote data collection is desired.

PROGRAM MDAP85

The maximum RAM available for the HP-85 is 32K, which is insufficient to hold the entire MDAP85 program in memory. Therefore, the program has been divided into eight integrated program modules and two independent utility modules. The program modules share a parameter file, data files, and program tape, and they automatically load and run each other. The utility modules are actually a separate set of programs designed for special applications and are not required for day-to-day data acquisition. A program flow chart showing the relationships between program modules, data files, Micromux hardware, and mainframe computer is given in figure B-3.

The program modules include:

1. Autost: This module prompts the operator to set time and date for use by several other program modules and is run automatically if the program tape is installed in the HP-85 on power up.
2. CONTRL: Main module for activating other modules.
3. INPUT: For entering instrument calibration factors and system operation parameters into file PFILE.
4. ACQUIR: Collects data from the Micromux receiver via RS232 interface, reduces and prints the data, and loads the scan into the data file.
5. PQUIR: Collects data from the Micromux receiver via telephone lines, reduces and prints the data, and loads the scan into the data file.
6. STORE: Transfers the program tape data file onto mass storage tapes.
7. PRINT: Prints the acquired data files.
8. PLOT: Plots a single data file on the CRT or graphics plotter.

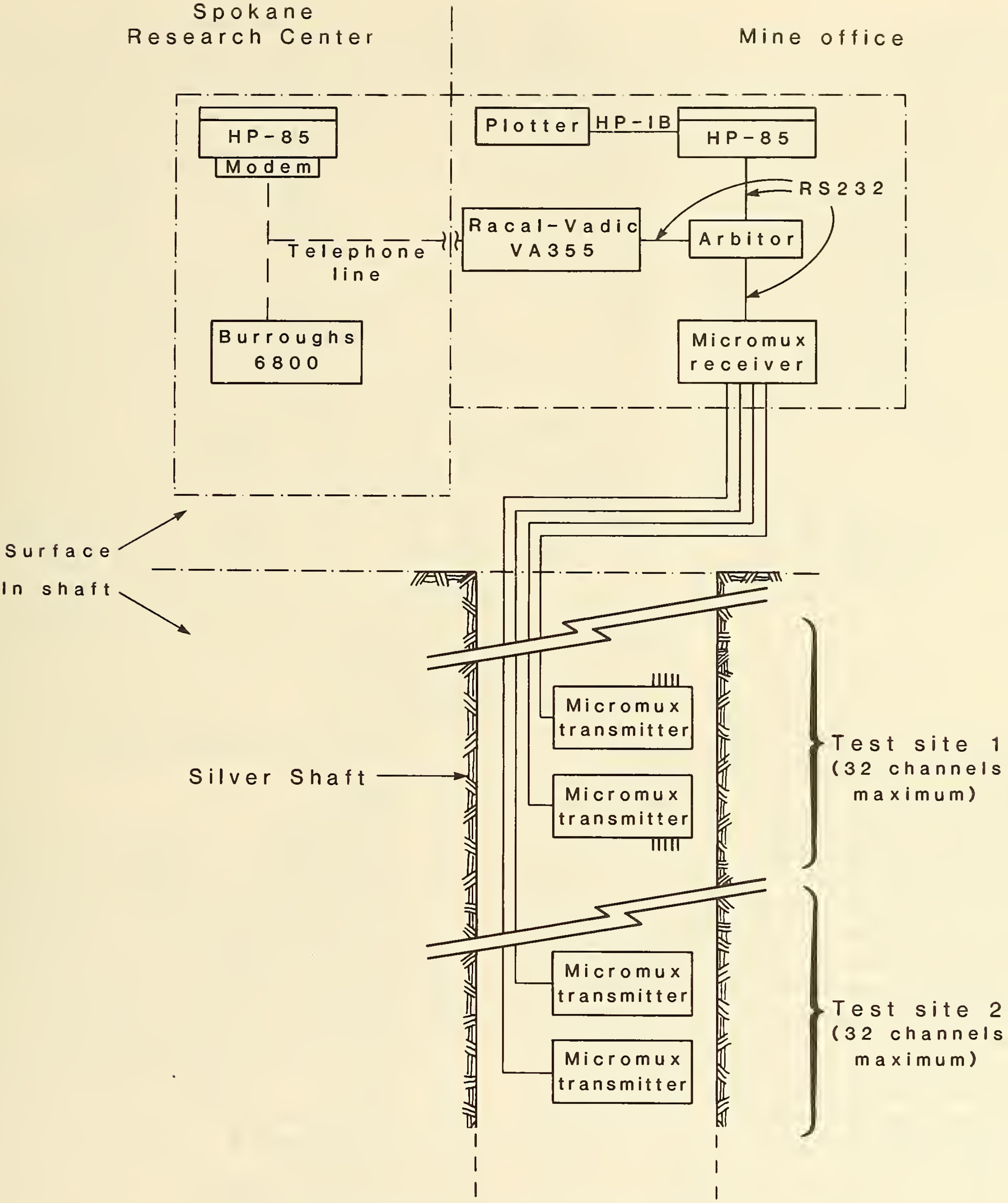


FIGURE B-1. - System configuration.

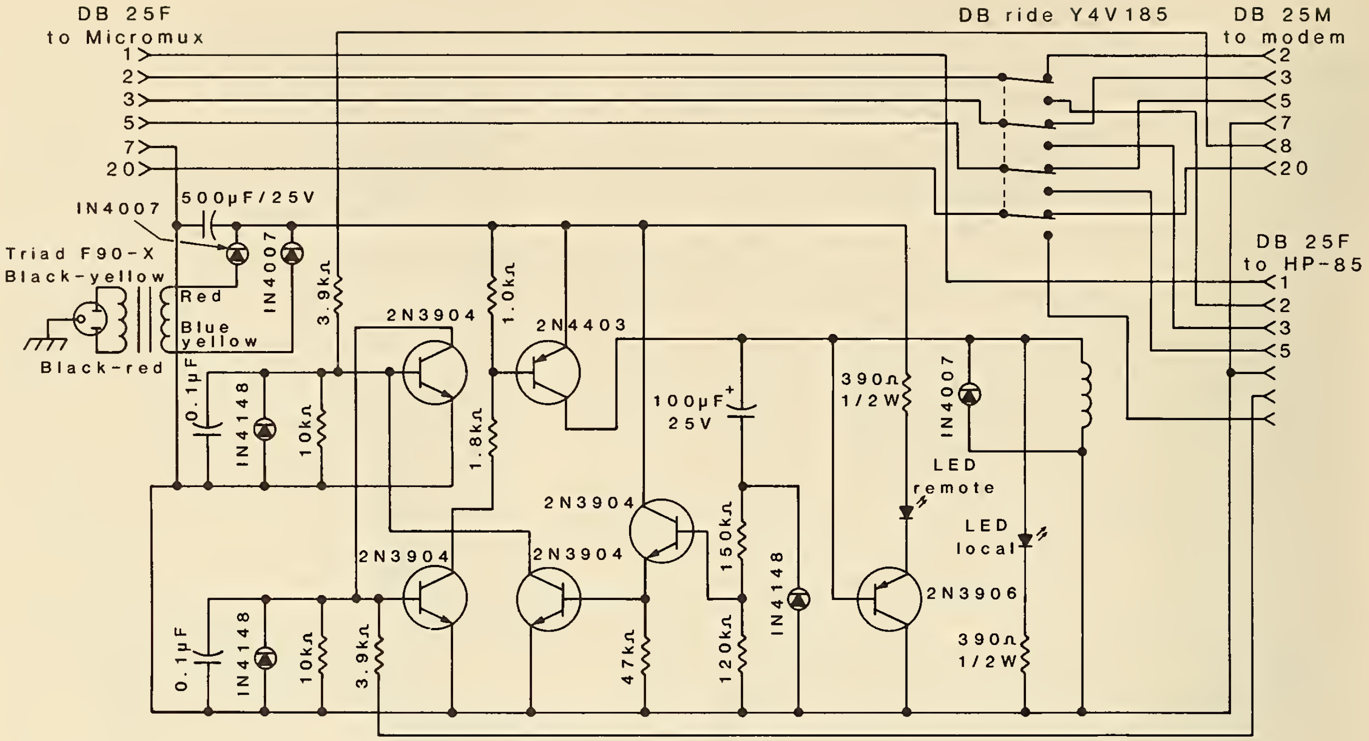


FIGURE B-2. - Arbiter schematic.

TABLE B-1. - Hardware list

Underground subsystem: Transmitters...........	Burr-Brown Micromux #6820
On-site data acquisition hardware:	
Microcomputer..................................	HP-85
Memory module (16K)........................	HP-82903A
Series 80 ROM drawer........................	HP-82936A
HP-IB interface................................	HP-82937A
Opt-001 serial interface....................	HP-82939A
HP-85 plotter-printer ROM..................	HP-00085-15002
HP-85 input-output ROM......................	HP-00085-15003
HP-85 advanced programming ROM.............	HP-00085-15005
Opt-002 graphics plotter......................	HP-7470A
Receiver..	Burr-Brown Micromux #6800
Hardware for remote data acquisition:	
Microcomputer..................................	HP-85
Memory module (16K)........................	HP-82903A
Series 80 ROM drawer........................	HP-82936A
HP-IB interface................................	HP-82937A
HP-85 plotter-printer ROM..................	HP-00085-15002
HP-85 input-output ROM......................	HP-00085-15003
HP-85 advanced programming ROM.............	HP-00085-15005
HP-95 modem......................................	HP-82950A
Opt-002 graphics plotter......................	HP-7470A
Receiver (on-site)............................	Burr-Brown Micromux #6800
Modem (on-site)................................	Racal-Vadic VA355
Additional hardware for combined remote and local data acquisition:[1] Arbiter (on-site)....	Custom built (figure B-2 for spec.)

[1] Only 1 receiver required for combined remote and local data acquisition.

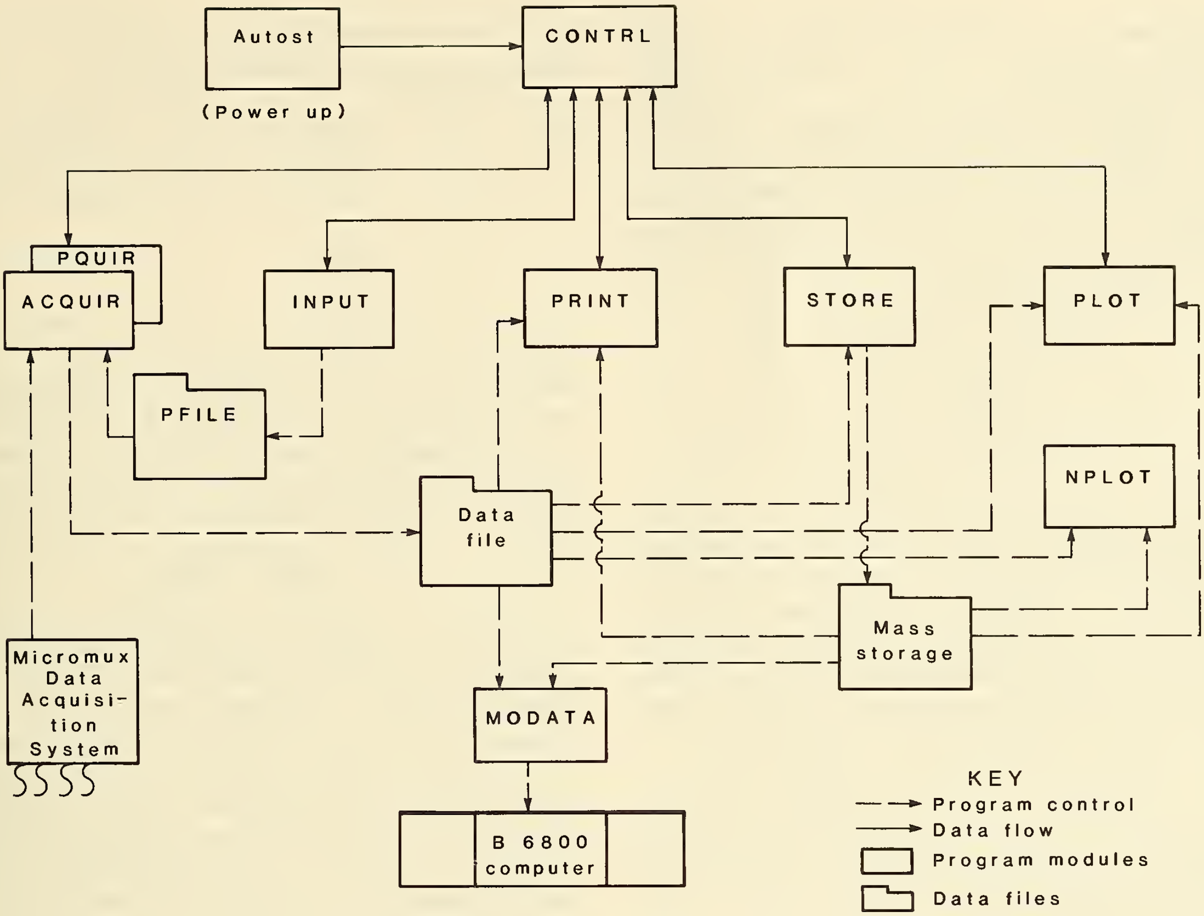


FIGURE B-3. - Program flow chart.

The utility modules include:

1. NPLOT: Plots multiple data files on the CRT or graphics plotter.

2. MODATA: Transmits data files as character data through the phone modem. This module was designed for uploading data to Spokane Research Center's Burroughs 6800 computer and may require modification for use with other computers.

The data files include:

1. PFILE: The parameter file that holds data reduction and program control parameters.

2. Data file: The file on the program tape that stores data as collected up to the maximum allowable number of scans.

3. Mass storage: A separate tape for permanent storage of completed data files.

Each of these modules, when activated, presents a menu on the CRT that shows the function of software keys (K1-K8) throughout that module. Then the operator can choose one of the options presented in the menu and press the corresponding software key in order to execute the option. In almost all cases, once execution of an option is completed, the program will again present the menu for a new option (exceptions being options to exit or stop a module).

For each program module, the menu always includes two options: HELP and EXIT (or STOP in the case of module CONTRL). The option HELP gives a brief description of the functions of software keys. In some modules, it also provides additional information regarding the execution of the program.

The EXIT option stops the execution of an active module and activates the CONTRL module automatically. This is the only way through which the operator can halt execution of one module and activate another one. In CONTRL module, the STOP option halts execution of program MDAP85 altogether.

The MDAP85 program can be activated either from the module Autost or by direct application of system LOAD and RUN commands to any of its eight program modules.

For example, enter:

```
LOAD "CONTRL"
```

and RUN.

MODULE AUTOST

The function of Autost is to set the computer's clock. The computer loses its settings of time and date any time the computer is turned off. Since the computer's time and date are recorded during data acquisition, they should be corrected when the power is turned back on again. Program Autost is started by:

1. Turning the power on while the program tape cartridge is installed in the tape drive; the system will automatically load and run Autost.
2. Inserting the program tape cartridge in the tape drive and typing the following commands:

```
Load "Autost"

RUN
```

Autost prompts the operator to enter the current time (24-hour format) and date.[2] After these values are entered satisfactorily, the program prompts whether or not the operator wants to run program MDAP85. Depending on the operator's answer, the program will either either activate the program module CONTRL or stop execution.

MODULE CONTRL

The function of this module is to activate the other modules of program MDAP85. When any of the other modules exit, control returns to module CONTRL. The key assignments for CONTRL are--

K1 - INPUT	K5 - PRINT
K2 - ACQUIR	K6 - PLOT
K3 - PQUIR	K7 - HELP
K4 - STORE	K8 - STOP

Notice that the first six software keys are assigned to six major modules of program MDAP85. The operator can activate any of these modules simply by pressing the corresponding key.

MODULE INPUT

The INPUT module allows the operator to enter instrument parameters and calibration factors into the computer for each channel. These parameters and factors are then written to a parameter file, PFILE, on the program tape cartridge, which is read by modules ACQUIR and PQUIR. The menu in this module presents the following assignments for the software keys:

K1 - GENERAL

General inputs include the address of the Micromux receiver, the number of Micromux transmitters, and the number of active "projects." A "project" is defined as a set of two transmitters (32 channels) that are monitored on the same time schedule. The number of channels per project can only be changed by modifying the program. A maximum of two projects may be monitored. The Silver Shaft system collects 32 channels of data from two Micromux transmitters at each instrumented level. The address of the Micromux receiver is zero. If a group of Micromux receivers are wired in the same RS232 line, each must have a unique address. See the Micromux manual for further details.

K2 - PROJECT

Project inputs are the project name, the excitation channel address, and the

[2] All of the alphabetic inputs and answers to the questions posed in module Autost or program MDAP85 must be in uppercase characters.

extensometer type. Instrumentation power supply voltage may vary and is included as a variable for data reduction. The extensometer type identifies the formula used in ACQUIR and PQUIR for extensometer data reductoion.

K3 - CHANNEL

Those parameters specific to each channel can be entered into the computer by this option. (Note that channels 1-32 are used for project 1 and channels 33-64 are used for project 2). The channel inputs consist of channel title, reduction type, printing flag, conversion factors, and low and high limits. The reduction type is an integer number that is used for two different purposes. First, it specifies the type of signal for each channel; this in turn identifies the type of formula to be used in module ACQUIR for conversion of signals to engineering units. Second, the reduction type is used to identify the individual extensometers and their anchors. The channel reduction types, their values, and the use of conversion factors is covered in the next section. The low and high limits specify the expected low and high values of the final converted data for each channel. These limits are used to show whether the data acquired are within the expected range.

K4 - PRINT 1

This option prints "PFILE" for project 1 (channels 1-32) on the HP-85 printer.

K5 - PRINT 2

This option prints "PFILE" for project 2 (channels 33-64).

K6 - HELP

K7 - EXIT

In the case of K1-K3, the computer displays the existing parameter values on the CRT. (The operator must enter a project number for PROJECT or a channel number for CHANNEL before the corresponding parameters are dislpayed.) In order to enter or change a parameter, the operator must roll the cursor to the appropriate location on the screen and change that parameter. After all parameters on the screen are satisfactory, "END LINE" is pressed and the computer reads the entire page of parameters.

When module INPUT is exited, the program writes all of the parameters in the file PFILE. To transfer PFILE and its contents from the program tape cartridge to another tape, module INPUT must be activated. When the menu is presented, the operator replaces the program tape cartridge with the second tape and executes the option EXIT. This procedure causes the parameter file, PFILE, to be transferred onto the second tape cartridge. The program MDAP85, however, stops execution after this operation.

MODULES ACQUIR AND PQUIR

These modules are used for data acquisition in program MDAP85. When activated, they read the contents of the parameter file written by module INPUT and create one active data file for each active project (if they do not already exist) on the program tape cartridge. These files are FILEØ1 and FILEØ2, which are used to store the data taken for projects 1 and 2, respectively. The menu includes the following software key assignments:

K1 - MANU 1

Each time this option is executed, the Micromux receiver will be prompted and data from channels 1-32 stored in FILEØ1. Then the scan (data taken from channels 1-32) will be printed on the HP-85 printer (fig. B-4) and the menu will be presented again.

K2 - MANU 2

This option is identical to MANU 1, except that the data will be taken for project 2 (channels 33-64) and the results will be written to FILEØ2.

K3 - AUTO 1

This option scans for project 1 data automatically. When this option is chosen, the operator will be prompted for the desired time interval between scans and also the time elapse before automatic data acquisition starts. The scan interval is the time between successive scans, and the time elapse is the time before the first data scan.

For example, if the time interval is 5 min and the time elapse is 20 min, the program waits for 20 min and then starts the process of taking data at 5-min intervals. The time interval can be any value between 3 and 1,440 min, while the time elapse can be any positive value up to 1,440 min.

After these values are entered, the program presents the menu and displays the time at which the next scan will be taken automatically. This routine is repeated after each scan. The actual process of taking, storing, and printing data by the computer in this mode is the same as in the manual mode.

While the program is waiting in the automatic mode, the operator can execute any other option presented in the menu. In order to stop or adjust the automatic data acquisition for project 1, the software key, K3, must be pressed while the menu is displayed on the CRT.

K4 - AUTO 2

This option takes data for project 2 in the automatic mode and is identical to AUTO 1.

K5 - DISP 1

This option takes data for project 1 and displays it on the CRT. Unlike the first four options, however, it does not store and print the data. After the data point is taken, the operator will be required to enter a particular channel number (for selective channels) or press "END LINE" (for successive channels) in order to display the data for each channel on the CRT. If a zero is entered instead of a channel number, the program will exit this option and present the menu.

K6 - DISP 2

This option is identical to DISP 1 except that it takes data for project 2.

K7 - HELP

K8 - EXIT

If automatic data acquisition is to be activated for both projects, the time interval and the time elapse for each project must be chosen carefully in order to avoid any time conflict. The operator should note that it might take up to 3 min per scan for data to be taken, stored, and printed. Therefore, the time difference between any two scans from projects 1 and 2 must be greater than 3 min to avoid conflict.

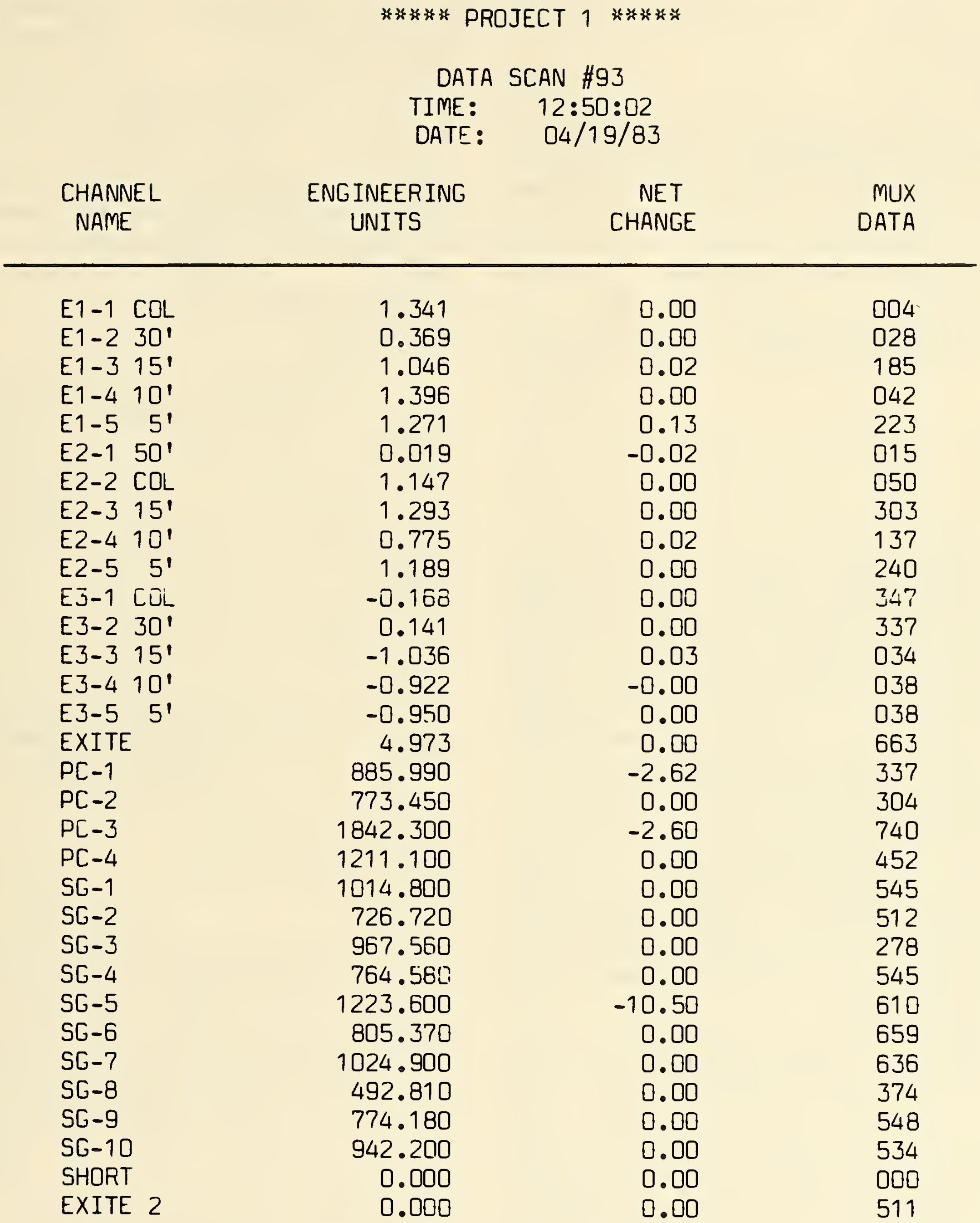
It was mentioned earlier that the data taken for projects 1 and 2 are stored in FILEØ1 and FILEØ2, respectively. Each of these files has a maximum storage capacity of 100 scans. These files must be transferred onto a mass storage tape before this capacity has been reached or the computer will halt. The transfer can be accomplished by using module STORE, as described in the next section.

When using the PQUIR version of this module, the phone number is inserted at line 298. The phone number consists of a string of digits and "@" characters. The @ character causes a second delay in the dialing sequence. This is convenient when dialing through building switchboards, etc., where a second dial tone is created.

Output from PQUIR differs from output from ACQUIR in that the raw data string is printed out before each data table in order to help identify line noise and protocol problems.

MODULE STORE

The data stored in files FILEØ1 and FILEØ2 can be transferred from the program tape cartridge to a mass storage tape by module STORE. For this module,



```
                        ***** PROJECT 1 *****

                           DATA SCAN #93
                         TIME:     12:50:02
                          DATE:     04/19/83

     CHANNEL          ENGINEERING          NET              MUX
      NAME               UNITS            CHANGE            DATA
  ________________________________________________________________

     E1-1 COL             1.341             0.00             004
     E1-2 30'             0.369             0.00             028
     E1-3 15'             1.046             0.02             185
     E1-4 10'             1.396             0.00             042
     E1-5  5'             1.271             0.13             223
     E2-1 50'             0.019            -0.02             015
     E2-2 COL             1.147             0.00             050
     E2-3 15'             1.293             0.00             303
     E2-4 10'             0.775             0.02             137
     E2-5  5'             1.189             0.00             240
     E3-1 COL            -0.168             0.00             347
     E3-2 30'             0.141             0.00             337
     E3-3 15'            -1.036             0.03             034
     E3-4 10'            -0.922            -0.00             038
     E3-5  5'            -0.950             0.00             038
     EXITE                4.973             0.00             663
     PC-1               885.990            -2.62             337
     PC-2               773.450             0.00             304
     PC-3              1842.300            -2.60             740
     PC-4              1211.100             0.00             452
     SG-1              1014.800             0.00             545
     SG-2               726.720             0.00             512
     SG-3               967.560             0.00             278
     SG-4               764.560             0.00             545
     SG-5              1223.600           -10.50             610
     SG-6               805.370             0.00             659
     SG-7              1024.900             0.00             636
     SG-8               492.810             0.00             374
     SG-9               774.180             0.00             548
     SG-10              942.200             0.00             534
     SHORT                0.000             0.00             000
     EXITE 2              0.000             0.00             511
```

FIGURE B-4. - 'ACQUIR' printed output.

the menu presents the following software key assignments:

K1 - STORE 1

This option transfers the data contained in FILEØ1 onto a mass storage tape.

K2 - STORE 2

This option transfers the data contained in FILEØ2 onto a mass storage tape.

K3 - HELP

K4 - EXIT

When either of the first two options is chosen, the program reads the data from the corresponding file into the internal memory of the computer. It then prompts the operator to replace the program tape cartridge with a mass storage tape. After this replacement, the program prints out the tape directory and prompts for a new file-name for the file being transferred. The file-name entered must have from one to six alpha-numeric characters, and must be different from those already in the tape directory. The program then transfers the data from the computer internal memory into the newly created mass storage tape file.

When the transfer is completed, the updated catalog will be presented and the operator will be asked to put the program tape cartridge back into the tape drive. At this point, the operator will be given a chance to purge the original file (whether the original is FILEØ1 or FILEØ2). The option for purging must be chosen if the file is full or if the operator wants to restart data acquisition with scan 1.

One of the most probable errors during data transfer occurs when the mass storage tape is full. In this case, the computer prints an error message on the thermal printer and gives the operator a chance to use amother mass storage tape cartridge.

This module can also be used to transfer files between mass storage tapes by renaming the file to be transferred as FILEØ1 or FILEØ2 and using the appropriate option.

MODULE PRINT

This module prints the contents of FILEØ1, FILEØ2, and mass storage files on the HP-85 thermal printer. In this module, the following software key assignments are in effect:

K1 - PRINT 1

This option prints the contents of FILEØ1.

K2 - PRINT 2

This option prints the contents of FILEØ2.

K3 - STORED

This option prints the contents of any file that has been stored on a mass storage tape by module STORE.

K4 - HELP

K5 - EXIT

If the option for printing a stored file is chosen, the program prompts the operator to insert the mass storage tape containing the stored data into the tape drive. Then it prints the tape directory and prompts for the name of the file to be printed. Note that the file-name entered must be one of those in the tape directory; otherwise, the program will print an error message and ask the operator to put the program tape cartridge back into the tape drive.

For all the print options, the program reads the file header and displays (on the CRT) the project number and ID, the total number of data scans in the file, the time and date of the first data scan, and the time and date of the last data scan stored in the file. Then it prompts for the time interval (starting and ending date and time) that identifies those data scans to be printed. After these values are entered, the program starts printing the data.

The printout contains the project file and information (project number and ID, etc.) and also all of the data scans taken within the time interval entered. For each data scan, the printout contains the data scan number, the time and date at which it was taken, and 32 channels of data.

For each channel, the channel number and title, raw and converted values, and the channel condition are printed out. The channel condition can indicate one of the following conditions: low (LOW), high (HIG), error (ERR), and change (CHG). The LOW and HIG indicate whether the data taken are lower or higher than expected limits. The ERR indicates that an error has occurred in the conversion of raw data. The CHG indicates that there has been a significant change (more than 5%) in the value of converted data, with respect to the average of values from the previous four data points. Figure B-5 shows a typical printed output from the "PRINT" option.

When the printing is finished, the program asks the operator to put the program tape back into the tape drive (if the option STORED was chosen for printing) and presents the menu.

MODULES PLOT AND NPLOT

These modules plot the converted data versus time on the CRT or the graphics plotter. The menu presents the following software key assignments:

K1 - PLOT 1

This option plots data from FILEØ1.

K2 - PLOT 2

This option plots data from FILEØ2.

K3 - STORED

This option plots data from any file that has been stored on a mass storage tape by module STORE.

K4 - HELP

K5 - EXIT

When changing tapes (for stored files) and reading the files, these software keys function in a manner similar to those in module PRINT. The only difference is when one of the plot options is chosen, the program transfers the contents of the corresponding file into the internal memory of the computer. This will serve to speed up the plotting process, since the program will not have to read the data from a file during plotting.

Whe the transfer of data is complete, the program starts the plotting process. First, it displays the project and file information on the CRT and prompts for the time interval (starting and ending date and time) for the plot; this time interval is used both for identification of the data to be plotted and for plotting and labeling the x-axis. The program also prompts for the minimum and maximum values to be used in plotting and labeling the y-axis.

After the above values are entered, the operator must choose between plotting on the CRT or the graphics plotter. Due to its size, the plots made on the CRT are very coarse and should be used only for a preliminary check of the results. The graphics plotter is used for making complete, detailed plots. If the operator chooses to plot on the CRT, the program will start plotting the x- and y-axis. Otherwise, it will ask for the y-axis label to be entered, and the plotter to be set (change of paper, etc.); then plotting and axes labeling is initiated on the graphics plotter using pen 1.

Next, the program asks whether to plot data. This question provides an opportunity to discontinue a plot; it also provides a pause for changing the plotter pen (in the case of the graphics plotter). If the operator decides to continue plotting, the program will ask for the following inputs: channel number for the plot, number of points to bypass for identification (graphics plotter only), option for connecting the data points (graphics plotter only), and a plotting factor or modulus for strain gauge channels (graphics plotter only).

```
                         ***** PROJECT 1 *****

                             DATA SCAN #1
                           TIME:  22:33:11
                           DATE:  02/19/83

CHANNEL        CHANNEL           CHANNEL          RAW         CON
 NUMBER          NAME             VALUE           DATA

-----------------------------------------------------------------------

    1          E1-1 50'        -6.870E-01          507         LOW
    2          E1-2 30'         1.956E-02          428         HIG
    3          E1-3 15'        -5.207E-02          417         LOW
    4          E1-4 10'        -6.868E-01          028         LOW
    5          E1-5  5'        -3.285E-01          331         LOW
    6          E2-1 50'        -1.318E+00          002         LOW
    7          E2-2 30'        -1.378E-01          372         LOW
    8          E2-3 15'        -6.021E-02          286         LOW
    9          E2-4 10'        -9.835E-01          018         LOW
   10          E2-5  5'        -1.205E-01          234         LOW
   11          E3-1 50'        -1.149E-01          334         LOW
   12          E3-2 30'        -4.124E-02          278         LOW
   13          E3-3 15'        -4.830E-02          265         LOW
   14          E3-4 10'        -6.206E-02          242         LOW
   15          E3-5  5'        -7.013E-02          248         LOW
   16          EXCITE           4.973E+00          663         HIG
   17          PC-1             9.192E+01          000         HIG
   18          PC-2             4.763E+01          000         HIG
   19          PC-3             7.220E+01          000         HIG
   20          PC-4             3.423E+01          000         HIG
   21          SG-1            -5.127E-06          000         LOW
   22          SG-2            -1.081E-05          000         LOW
   23          SG-3            -1.486E-05          000         LOW
   24          SG-4            -5.481E-05          000         LOW
   25          SG-5            -1.978E-05          000         LOW
   26          SG-6            -7.898E-06          000         LOW
   27          SG-7            -6.756E-06          000         LOW
   28          SG-8            -7.515E-06          000         LOW
   29          SG-9            -7.630E-05          000         LOW
   30          SG-10           -4.246E-05          000         LOW
   31                           0.000E+00          000
   32                           0.000E+00          505
```

FIGURE B-5. - 'PRINT' output format.

Then, it plots the data for the channel number entered.

The process of plotting data (described above) will be repeated until the operator discontinues plotting, or the data for a maximum of six channels are plotted. After the plotting is discontinued, the operator will be asked if more plots are to be made from the file currently in computer memory. Depending on the answer, the program will either restart the plotting process or display the menu.

Module PLOT has the capacity of plotting data from only one file per graph. In order to plot data from several files, the utility program NPLOT can be independently loaded and activated. With a few exceptions, the inputs and outputs of program NPLOT are similar to those of module PLOT. One such exception is that it is not necessary to remove the mass storage tapes and replace the program tape until all desired files are plotted.

MODULE MODATA

This module is a modified version of HP's MODCOM, which is supplied with the 82950A modem. It has been modified to translate numeric data files written by MDAP85 with 32 channels of data to character data and transmit those data to the SRC B6800 computer. A sequential data file with automatic line numbering on each carriage return must be set up on the B6800 to receive the data. There is no host prompt, and no delay is required between lines. Specific software instructions are the same as for the original MODCOM program and will not be repeated here. A binary program supplied with MODCOM is necessary to run MODATA.

CONVERSION OF DATA TO ENGINEERING UNITS

During data acquisition, the Micromux telemetry equipment converts the analog signals for each channel to integer values (raw data) ranging from 000 to 999. The raw data, when received by HP-85, must be converted to engineering units. This conversion takes place in module ACQUIR using the parameters and conversion factor stored in parameter file PFILE.

First, the raw data for each channel are converted to voltage according to the following formula:

$$V_i = D_i \cdot C\emptyset_i - C1_i$$

where i = channel number

C∅ = conversion factor

C1 = voltage offset

D = raw data

V = channel voltage

After this conversion, various reduction formulas are used to convert the voltage to other engineering units. The specific formula used for each channel depends on its signal type, which is determined from its reduction type entered in module INPUT.

For the current system setup, five types of signals are analyzed. These signal types are described in the following sections.

Excitation Voltage

For this type of signal, no other conversion takes place after the conversion of raw data to voltage. The reduction type for a channel with this type of signal (excitation channel) must be set to zero. The excitation voltage is used in the conversion of data for other types of signals.

Borehole Extensometers

The reduction type for a channel with this type of signal is computed according to the following formula:

$$R_i = 10N + r$$

where i = channel number

N = extensometer number

r = anchor number

R = reduction type.

In this way, the individual extensometers and anchors can be identified by the channel reduction type. The reference anchor for each extensometer type must be set equal to one. This program is currently set up for five anchor extensometers.

The extensometer type (referred to in module INPUT) specifies the formula used for data reduction. The Silver Shaft project used a reduction type set equal to two, which specifies the following steps:

$$\Delta_i = C_2 D_i (V_{ex}/V_c) + C_3$$

where Δ_i = channel i's absolute displacement

C_2 = calibration slope (in/digit)

C_3 = calibration zero

D_i = digital output for channel i

V_{ex} = excitation voltage

V_c = calibration voltage

Solve for relative displacement from "stable" reference anchor:

for r = 1 $\quad d_i = \Delta_i$

for r = 2 to 5 $\quad d_i = d_1 - \Delta_i$

Concrete Pressure Cells

The reduction type for this type of signal must be set equal to four. The following formula is used for conversion of voltage to tangential linear stress for channel i:

$$\sigma_i = [V_i \cdot C3_i/(C5_i \cdot V_{ex})] + C4_i/C5_i + C2$$

where C2 = offset from calibration, psi, set at installation

C3 = voltage used in calibration

C4 = calibration offset, V

C5 = slope from calibration, V/psi

σ = pressure, psi.

Embedment Strain Gauges

The reduction type for this type of signal must be set equal to six. The following formula is used for conversion of voltage to tangential linear strains for channel i:

$$\varepsilon_i = [4V_i/[V_{ex} \cdot C3_i] - C4_i]$$

where C3 = gauge factor

C4 = strain offset

ε = linear strain.

Temperature Sensor

The reduction type for this type of signal must be set equal to one. The following formula is used for conversion of voltage into temperature for channel i:

$$T_i = C2_i \cdot V_i - C3_i$$

where C2 = conversion factor (°F/V)

C3 = temperature offset, °F

T = temperature, °F.

APPENDIX C.--PROGRAM LISTINGS

PROGRAM 'AUTOST'

```
100 ! This is the "Autost"
110 ! program for HP-85.
120 ! It asks for time and date
130 ! and runs program 'MDAP85'.
140 DIM R$[30]
150 ON ERROR GOTO 160
160 CLEAR
170 DISP USING 180
180 IMAGE "        PROGRAM 'Autost'",3/
190 DISP "Enter DATE and TIME"
200 LINPUT "mm/dd/yyyy hh:mm:ss",R$
210 R$=TRIM$(R$)
220 A=HMS(R$[LEN(R$)-7])
230 B=MDY(R$[1,10])-MDY("12/31/1981")
240 SETTIME A,B
250 ON ERROR GOTO 260
260 CLEAR
270 DISP "TIME: ",TIME$
280 DISP "DATE: ",MDY$(DATE+MDY("12/31/1981"))
290 DISP USING 300
300 IMAGE 3/
310 LINPUT "TIME, DATE ok(Y/N)?",A$
320 IF A$="N" THEN 160
330 IF A$#"Y" THEN 260
340 ON ERROR GOTO 400
350 CLEAR
360 LINPUT "Run program 'MDAP85'(Y/N)?",A$
370 IF A$="N" THEN 400
380 IF A$#"Y" THEN 350
390 CHAIN "CONTRL"
400 STOP
410 END
```

MODULE 'CONTRL'

```
100 ! This is module ’CONTRL’.
102 ! It activates all other
104 ! modules of prog. ’MDAP85’
106 ! defined by keys K1-K6.
108 GOSUB 136
110 ENABLE KBD 32
112 ON KEY# 1,"INPUT" GOSUB 156
114 ON KEY# 2,"ACQUIR" GOSUB 162
116 ON KEY# 3,"PQUIR" GOSUB 168
118 ON KEY# 4,"STORE" GOSUB 174
120 ON KEY# 5,"PRINT" GOSUB 180
122 ON KEY# 6,"PLOT " GOSUB 186
124 ON KEY# 7,"HELP " GOTO 192
126 ON KEY# 8,"STOP " GOTO 250
128 CLEAR @ KEY LABEL
130 DISP USING 132
132 IMAGE 3/,"   *****’CONTRL’ options*****"
134 GOTO 134
136 OFF KEY# 1
138 OFF KEY# 2
140 OFF KEY# 3
142 OFF KEY# 4
144 OFF KEY# 5
146 OFF KEY# 6
148 OFF KEY# 7
150 OFF KEY# 8
152 ENABLE KBD 255
154 RETURN
156 GOSUB 136
158 CHAIN "INPUT"
160 RETURN
162 GOSUB 136
164 CHAIN "ACQUIR"
166 RETURN
168 GOSUB 136
170 CHAIN "PQUIR"
172 RETURN
174 GOSUB 136
176 CHAIN "STORE"
178 RETURN
180 GOSUB 136
182 CHAIN "PRINT"
184 RETURN
186 GOSUB 136
188 CHAIN "PLOT"
190 RETURN
192 GOSUB 136
194 CLEAR @ ENABLE KBD 1
196 DISP USING 198
198 IMAGE 9X,"*****HELP*****",2/
200 DISP "The keys operate as follows:"
```

```
202 DISP USING 204
204 IMAGE /,"K1= run module 'INPUT'; it is",/,"the segment for enterin
g"
206 DISP USING 208
208 IMAGE "test information and channel",/,"conversion factors."
210 DISP USING 212
212 IMAGE /,"K2= run module 'ACQUIR'; it is",/,"the segment for data",
/,"acquisition.",3/
214 LINPUT "Hit 'END LINE' for more help",A$
216 CLEAR
218 DISP USING 220
220 IMAGE /,"K3= run module 'PQUIR'; it is",/,"the segment for phone d
ata",/,"acquisition.",2/
224 DISP USING 226
226 IMAGE "K4= run module 'STORE'; it",/,"reads the data files created
 by"
228 DISP USING 230
230 IMAGE "module 'ACQUIR' and stores",/,"them in a mass storage tape.
",3/
233 LINPUT "Hit 'END LINE' for more help",A$
234 CLEAR
235 DISP USING 236
236 IMAGE /, "K5= run module 'PRINT'; it",/,"prints out the test resul
ts."
238 DISP USING 240
240 IMAGE /,"K6= run module 'PLOT'; it",/,"plots the test results."
242 DISP USING 244
244 IMAGE /,"K7= 'HELP'.",2/,"K8= 'STOP' program 'MDAP85'.",2/
246 LINPUT "Hit 'END LINE' to return",A$
248 GOTO 108
250 GOSUB 136
252 CLEAR
254 STOP
256 END
```

MODULE 'PQUIR'

```
10 ! This is module 'PQUIR'.
20 ! It is the modem version
30 ! of 'ACQUIR' for remote
40 ! use of program 'MDAP85'.
50 ! -----PHONE VERSION------
60 OPTION BASE 1
70 DIM P$[64],H$[512],B$[204]
80 DIM K$[30],K1$[1]
90 INTEGER M,T,N,E(2),X(2),R(64),S(64),D0,D1(2),D2,D(64),Y(2),I,J,K,K1
,Q,Q1,L1(2),L2(2),G,O
100 INTEGER Z(3,6),J1,J2,D3(32),L3(2,16),L4(2,16),A(2)
110 SHORT C(9,64),V(64),F(64),F1(32),Z1(6),V1,F2(32)
120 REAL T0,T1(2),T2,T3,T4,T5,T6,T7(2),T8(2),T9,U
130 D2=DATE @ T2=TIME
140 CLEAR
150 DISP USING 160
160 IMAGE 9X,"MODULE 'PQUIR'",2/
170 DISP "TIME: ",HMS$(T2)
180 DISP "DATE: ",MDY$(D2+MDY("12/31/1981"))
190 DISP USING 200
200 IMAGE 3/,".....Wait"
210 WAIT 2000 @ ENABLE KBD 0
220 !
230 ! Read parameter file.
240 ASSIGN# 1 TO "PFILE"
250 CRT OFF
260 READ# 1 ; N,M,T,E(),X(),R(),S(),C(,),P$,H$
270 CLEAR
280 ASSIGN# 1 TO *
290 !
300 ! Initialize variables.
310 D1(1)=D2 @ D1(2)=D2
320 T1(1)=T2 @ T1(2)=T2
330 Y(1)=0 @ Y(2)=0
340 A(1)=0 @ A(2)=0
350 FOR I=1 TO 6
360 Z1(I)=0
370 FOR J=1 TO 3
380 Z(J,I)=0
390 NEXT J
400 NEXT I
410 FOR J=1 TO 32
420 F1(J)=0
430 NEXT J
440 ! Create output files.
450 ON ERROR GOTO 510
460 ASSIGN# 1 TO "FILE01"
470 OFF ERROR
480 PRINT# 1,1 ; P$[1,30],H$[1,256]
490 READ# 1,2 ; Y(1),D1(1),T1(1)
500 GOTO 570
510 OFF ERROR
```

```
520 CREATE "FILE01",102,586
530 ASSIGN# 1 TO "FILE01"
540 CRT OFF
550 PRINT# 1,1 ; P$[1,30],H$[1,256]
560 PRINT# 1,2 ; Y(1),D1(1),T1(1),D2,T2,1
570 IF N=1 THEN 710
580 ON ERROR GOTO 650
590 ASSIGN# 2 TO "FILE02"
600 OFF ERROR
610 PRINT# 2,1 ; P$[31,60],H$[257,512]
620 READ# 2,2 ; Y(2),D1(2),T1(2)
630 OFF ERROR
640 GOTO 730
650 OFF ERROR
660 CREATE "FILE02",102,586
670 ASSIGN# 2 TO "FILE02"
680 CRT OFF
690 PRINT# 2,1 ; P$[31,60],H$[257,512]
700 PRINT# 2,2 ; Y(2),D1(2),T1(2),D2,T2,2
710 !
720 ! Key assignments.
730 ENABLE KBD 32
740 ON KEY# 1,"MANU 1" GOTO 2600
750 ON KEY# 2,"MANU 2" GOTO 2640
760 ON KEY# 3,"AUTO 1" GOTO 3110
770 ON KEY# 4,"AUTO 2" GOTO 3130
780 ON KEY# 5,"DISP 1" GOTO 3390
790 ON KEY# 6,"DISP 2" GOTO 3410
800 ON KEY# 7,"HELP  " GOTO 3820
810 ON KEY# 8,"EXIT  " GOTO 4040
820 CLEAR @ KEY LABEL
830 DISP USING 840
840 IMAGE 3/,"    ***** 'PQUIR' options*****",2/
850 DISP USING 860 ; Y(1),Y(2)
860 IMAGE "Proj #1: ",DDD," scan(s)"/"Proj #2: ",DDD, " scan(s)",2/
870 FOR I=1 TO 2
880 IF A(I)=0 THEN 910
890 DISP USING 900 ; I,HMS$(T7(I))
900 IMAGE "AUTO ",D," active at  ",K
910 NEXT I
920 GOTO 920
930 !
940 ! Call to 'MICROMUX'.
950 ! ------OVER PHONE LINE-----
960 ENABLE KBD 255
970 S=10 @ M=0 @ T=2 @ X=0
980 O=0
990 IOBUFFER B$ @ CONTROL B$,0 ; 1 ! USABLE 196
1000 K$="9@1@2087441459"
1010 CONTROL S,4 ; 26
1020 CONTROL S,2 ; 6
1030 WAIT 2000 ! WAIT FOR DIAL TONE
```

```
1040 FOR I=1 TO LEN(K$)
1050 K1$=K$[I,I]
1060 IF K1$="@" THEN WAIT 500 @ GOTO 1150
1070 K=NUM(K1$)-48
1080 IF K=0 THEN K=10
1090 FOR J=1 TO K
1100 ASSERT S;5 @ ON TIMER# 3,51 GOTO 1120
1110 GOTO 1110
1120 ASSERT S;6 @ ON TIMER# 3,25 GOTO 1140
1130 GOTO 1130
1140 OFF TIMER# 3 @ NEXT J
1150 WAIT 700 @ NEXT I
1160 ON TIMER# 3,30000 GOTO 1180
1170 GOTO 1190
1180 CONTROL 10,2 ; 4@ WAIT 30000 @ PRINT "1180 ERR" @ OFF TIMER# 3 @
GOTO 950
1187 HALT 10 @ OFF TIMER# 3 @ GOTO 950
1190 CONTROL S,2 ; 7
1200 STATUS 10,3 ; B@ Z4=0
1210 IF B=1 THEN 1230 ELSE B=B-2
1220 IF B<1 THEN 1200 ELSE 1210
1230 OFF TIMER# 3
1231 ON ERROR GOTO 1233
1232 B$="0" @ TRANSFER B$ TO 10 INTR
1233 ON ERROR GOTO 1232
1235 CONTROL B$,0 ; 1,0
1240 B$="" @ OUTPUT 10 USING "#,Z,A" ; M,"?"
1250 TRANSFER 10 TO B$ INTR ; COUNT T*49 @ WAIT 5000
1255 OFF ERROR
1260 CONTROL 10,2 ; 4
1270 PRINT B$
1280 TO=TIME
1285 ON TIMER# 3,30000 GOTO 1187
1290 OFF TIMER# 3 @ DO=DATE
1291 IF NUM(B$)<48 THEN PRINT "1291" @ ENTER B$ USING "#,X" @ X=1 @ SF
LAG 10
1292 IF NUM(B$)>57 THEN PRINT "1292" @ ENTER B$ USING "#,X" @ X=1 @ SF
LAG 10
1300 FOR I=T TO 1 STEP -1
1310 K=16*(I-1)
1320 FOR J=1 TO 16
1330 ENTER B$ USING "#,3Z" ; D(J+K)
1340 NEXT J
1350 IF X<2 THEN PRINT "1350" @ ENTER B$ USING "#,X"
1355 X=X+1
1360 NEXT I
1365 IF FLAG(10) THEN B$="" @ TRANSFER 10 TO B$ INTR ; COUNT 1
1370 RETURN
1380 !
1390 ! Change to engin. unit.
1400 L1(1)=-1 @ L1(2)=-1
1410 L2(1)=-1 @ L2(2)=-1
```

```
1420 FOR I=1 TO 16
1430 L3(1,I)=0 @ L3(2,I)=0
1440 L4(1,I)=0 @ L4(2,I)=0
1450 NEXT I
1460 J=E(G)
1470 ON ERROR GOTO 1570
1480 V(J)=C(1,J)*D(J)+C(2,J)
1490 F(J)=V(J)
1500 V1=V(J)
1510 K=IP(J/17)+1
1520 IF G=2 THEN K=IP((J-32)/17)+1
1530 K1=J-16*K-32*G+48
1540 IF F(J)<C(8,J) THEN L3(K,K1)=1
1550 IF F(J)>C(9,J) THEN L4(K,K1)=1
1560 GOTO 1590
1570 PRINT USING 1580 ; G
1580 IMAGE "*****ERROR - CONVERSION OF DATA",/,"EXCITATION CHANNEL, PR
OJECT",3D
1590 FOR I=32*(G-1)+1 TO 32*G
1600 ON ERROR GOTO 1990
1610 IF I=J THEN 2070
1620 V(I)=C(1,I)*D(I)+C(2,I)
1630 Q=IP(I/17)+1
1640 IF G=2 THEN Q=IP((I-32)/17)+1
1650 Q1=I-16*Q-32*G+48
1660 IF R(I)<11 THEN 1740
1670 J1=IP(R(I)/10)
1680 J2=RMD(R(I),10)
1690 IF J1>3 THEN 1920
1700 IF J2<1 THEN 1920
1710 IF J2>6 THEN 1920
1720 Z(J1,J2)=I
1730 ON X(G) GOTO 1790,1860
1740 ON R(I)+1 GOTO 1750,1770,1790,1810,1840,1860,1900,1920,1920,1920,
1920
1750 F(I)=V(I)
1760 GOTO 1950
1770 F(I)=C(3,I)*V(I)-C(4,I)
1780 GOTO 1950
1790 F(I)=C(4,I)*V(I)/(C(6,I)*V1)-C(5,I)/C(6,I)
1800 GOTO 1950
1810 U=LOG((V1/V(I)-1)*C(4,I))
1820 F(I)=1/(C(3,I)*U+C(4,I))-C(5,I)
1830 GOTO 1950
1840 F(I)=C(4,I)*V(I)/(C(6,I)*V1)+C(5,I)/C(6,I)+C(3,I)
1850 GOTO 1950
1860 F(I)=C(3,I)*D(I)*V1/5.0075+C(4,I)
1870 ! NEW EXTENSOMETER
1880 ! REDUCTION FORMULA
1890 GOTO 1950
1900 F(I)=4*V(I)/(V1*C(4,I))-C(5,I)
1910 GOTO 1950
```

```
1920 V(I)=0
1930 F(I)=0
1940 GOTO 2070
1950 IF R(I)>10 THEN 2070
1960 IF F(I)<C(8,I) THEN L3(Q,Q1)=1
1970 IF F(I)>C(9,I) THEN L4(Q,Q1)=1
1980 GOTO 2070
1990 IF O=1 THEN 2020
2000 O=1 @ BEEP @ PRINT USING 2010 ; Y(G)+1
2010 IMAGE "*****ERROR - CONVERSION OF SCAN:",/,DDD,"   CHANNEL(S)"
2020 PRINT USING 2030 ; I
2030 IMAGE 15X,DDD
2040 V(I)=0 @ F(I)=0
2050 L3(Q,Q1)=1
2060 L4(Q,Q1)=1
2070 NEXT I
2080 FOR J1=1 TO 3
2090 ON ERROR GOTO 2110
2100 Z1(1)=F(Z(J1,1))
2110 FOR J2=2 TO 5
2120 ON ERROR GOTO 2140
2130 Z1(J2)=Z1(1)-F(Z(J1,J2))
2140 NEXT J2
2150 FOR J2=1 TO 5
2160 ON ERROR GOTO 2260
2170 I=Z(J1,J2)
2180 Z(J1,J2)=0
2190 F(I)=Z1(J2)
2200 Z1(J2)=0
2210 Q=IP(I/17)+1
2220 IF G=2 THEN Q=IP((I-32)/17)+1
2230 Q1=I-16*Q-32*G+48
2240 IF F(I)<C(8,I) THEN L3(Q,Q1)=1
2250 IF F(I)>C(9,I) THEN L4(Q,Q1)=1
2260 NEXT J2
2270 NEXT J1
2280 OFF ERROR
2290 FOR I=1 TO 2
2300 FOR J=1 TO 15
2310 IF L3(I,J)=0 THEN L1(I)=L1(I)-2^(J-1)
2320 IF L4(I,J)=0 THEN L2(I)=L2(I)-2^(J-1)
2330 NEXT J
2340 IF L3(I,16)=0 THEN L1(I)=BINEOR(L1(I),-32768)
2350 IF L4(I,16)=0 THEN L2(I)=BINEOR(L2(I),-32768)
2360 NEXT I
2370 RETURN
2380 !
2390 ! Write record on files.
2400 J1=32*(G-1)+1
2410 J2=32*G
2420 K=0
2430 FOR I=J1 TO J2
```

```
2440 K=K+1
2450 F2(I)=F(I)-F1(K)
2460 F1(K)=F(I)
2470 D3(K)=D(I)
2480 NEXT I
2490 IF Y(G)>1 THEN 2510
2500 D1(G)=D0 @ T1(G)=T0
2510 CRT OFF
2520 PRINT# G,2 ; Y(G),D1(G),T1(G),D0,T0,G
2530 PRINT# G,Y(G)+2 ; T0,D0,F1(),D3(),L1(),L2()
2540 CRT ON
2550 RETURN
2560 !
2570 ! Manu, Auto project # 1,2.
2580 ON TIMER# 1,T8(1)*60000 GOTO 2580
2590 T7(1)=RMD(TIME+T8(1)*60,86400)
2600 G=1
2610 GOTO 2650
2620 ON TIMER# 2,T8(2)*60000 GOTO 2620
2630 T7(2)=RMD(TIME+T8(2)*60,86400)
2640 G=2
2650 IF Y(G)<100 THEN 2700
2660 BEEP
2670 PRINT USING 2680 ; G
2680 IMAGE "*****ERROR - MAX # OF SCANS",/,"FOR PROJECT # ",D," HAS BE
EN",/,"REACHED."
2690 GOTO 730
2700 CLEAR @ ENABLE KBD 0
2710 GOSUB 970
2720 IF O#0 THEN 730
2730 GOSUB 1400
2740 Y(G)=Y(G)+1
2750 GOSUB 2400
2760 CRT OFF
2770 PRINT USING 2780 ; G
2780 IMAGE ,3/,5X,"*****PROJECT  ",D," *****"
2790 PRINT USING 2800 ; Y(G)
2800 IMAGE "SCAN # ",DDD
2810 PRINT "TIME: ",HMS$(T0)
2820 PRINT "DATE: ",MDY$(D0+MDY("12/31/1981"))
2830 PRINT USING 2840
2840 IMAGE 2/,"CHANNEL    ENGR.     NET    uMUXNAME       UNITS     CH
ANGE DATA"
2850 PRINT "--------------------------------"
2860 FOR I=21 TO 30
2870 F(I)=F(I)*1000000 @ F2(I)=F2(I)*1000000
2880 NEXT I
2890 FOR I=32*(G-1)+1 TO 32*G
2900 IF S(I)=0 THEN 3050
2910 Q=IP(I/17)+1
2920 IF G=2 THEN Q=IP((I-32)/17)+1
2930 Q1=I-16*Q-32*G+47
```

```
2940 K=8*(I-1)+1
2950 K1=K+7
2960 J=0
2970 IF BIT(L1(Q),Q1) THEN J=J+1
2980 IF BIT(L2(Q),Q1) THEN J=J+2
2990 IF J=0 THEN A$="   "
3000 IF J=1 THEN A$="LOW"
3010 IF J=2 THEN A$="HIG"
3020 IF J=3 THEN A$="ERR"
3030 PRINT USING 3040 ; H$[K,K1],F(I),F2(I),D(I)
3040 IMAGE 8A,X,MDDDDD.DDD, X,MDDD.DD,X,3Z
3050 NEXT I
3060 PRINT USING 3070
3070 IMAGE 5/
3080 GOTO 730
3090 !
3100 ! Auto, project # 1,2.
3110 G=1
3120 GOTO 3140
3130 G=2
3140 IF A(G)=0 THEN 3180
3150 A(G)=0
3160 OFF TIMER# G
3170 GOTO 730
3180 A(G)=1
3190 CLEAR
3200 ENABLE KBD 1
3210 ON ERROR GOTO 3190
3220 LINPUT "Input time interval between scans (MIN=3 min,MAX=1440 min
)",A$
3230 T3=VAL(A$)
3240 IF T3<3 THEN 3190
3250 IF T3>1440 THEN 3190
3260 LINPUT "Input time elapse befor start   (MIN=0 min,MAX=1440 min)"
,A$
3270 T9=VAL(A$)
3280 IF T9<0 THEN 3260
3290 IF T9>1440 THEN 3260
3300 IF T9=0 THEN T9=.0005
3310 OFF ERROR
3320 T8(G)=T3
3330 IF G=1 THEN ON TIMER# G,T9*60000 GOTO 2580
3340 IF G=2 THEN ON TIMER# G,T9*60000 GOTO 2620
3350 T7(G)=RMD(TIME+T9*60,86400)
3360 GOTO 730
3370 !
3380 ! Display of data.
3390 G=1
3400 GOTO 3420
3410 G=2
3420 CLEAR @ ENABLE KBD 255
3430 GOSUB 970
```

```
3440 IF O#0 THEN 730
3450 GOSUB 1400
3460 J1=32*(G-1)+1
3470 J2=32*G
3480 I=J1-1
3490 ENABLE KBD 1
3500 DISP "Input channel # to be displayed"
3510 DISP USING 3520 ; J1,J2
3520 IMAGE 2/,2X,DD,"-",DD,3X,"= channel for display",/,"'END LINE'= n
ext channel"
3530 DISP USING 3540
3540 IMAGE 4X,"0",5X,"= exit display",3/
3550 ON ERROR GOTO 3640
3560 LINPUT "Channel #:",A$
3570 J=VAL(A$)
3580 OFF ERROR
3590 IF J=0 THEN 730
3600 IF J<J1 THEN 3550
3610 IF J>J2 THEN 3550
3620 I=J
3630 GOTO 3670
3640 OFF ERROR
3650 I=I+1
3660 IF I>J2 THEN I=J1
3670 Q=IP(I/17)+1
3680 IF G=2 THEN Q=IP((I-32)/17)+1
3690 Q1=I-16*Q-32*G+47
3700 K=8*(I-1)+1
3710 K1=K+7
3720 CLEAR
3730 DISP USING 3740 ; I,H$[K,K1],F(I)
3740 IMAGE "Channel #",6X,"= ",DD,/,"Title",10X,"= ",K,/,"Channel valu
e",2X,"= ",MD.DDDe
3750 DISP USING 3760 ; D(I),V(I)
3760 IMAGE "Raw data",7X,"= ",3Z,/,"Voltage",8X,"= ",MDD.DDD
3770 DISP USING 3780 ; BIT(L1(Q),Q1),BIT(L2(Q),Q1)
3780 IMAGE "Low",12X,"= ",D,/,"High",11X,"= ",D,8/
3790 GOTO 3550
3800 !
3810 ! Help.
3820 ENABLE KBD 1 @ CLEAR
3830 DISP USING 3840
3840 IMAGE 9X,"*****HELP*****",2/
3850 DISP "The keys operate as follows:"
3860 DISP USING 3870
3870 IMAGE 1/,"K1= 'MANUAL' acquisition proj.1",/,"K2= 'MANUAL' acquis
ition proj.2"
3880 DISP USING 3890
3890 IMAGE "K3= 'AUTO' acquisition proj.1",/,"K4= 'AUTO' acquisition p
roj.2"
3900 DISP USING 3910
3910 IMAGE "K5= 'DISPLAY' data for proj.1",/,"K6= 'DISPLAY' data for p
```

```
roj.2"
3920 DISP USING 3930
3930 IMAGE "K7= 'HELP'",/,"K8= 'EXIT' module 'ACQUIR'",3/
3940 LINPUT "Hit 'END LINE' for more help",A$
3950 CLEAR
3960 DISP USING 3970
3970 IMAGE 2/,"Note: the max # of scans that can be stored for each pr
oj.     is 100."
3980 DISP USING 3990
3990 IMAGE "In 'DISPLAY' mode, the data",/,"taken will not be stored i
n",/,"files.",7/
4000 LINPUT "Hit 'END LINE' to return",A$
4010 GOTO 730
4020 !
4030 ! Exit.
4040 CLEAR @ ENABLE KBD 0
4050 OFF TIMER# 1 @ OFF TIMER# 2
4060 ASSIGN# 1 TO *
4070 ASSIGN# 2 TO *
4080 ENABLE KBD 33 @ CRT ON
4090 CHAIN "CONTRL"
4100 END
```

MODULE 'ACQUIR'

```
100 ! This is module 'ACQUIR'.
102 ! It is the acquiring seg-
104 ! ment of data acquisition
106 ! program 'MDAP85'.
108 OPTION BASE 1
110 DIM P$[64],H$[512],B$[204]
112 INTEGER M,T,N,E(2),X(2),R(64),S(64),D0,D1(2),D2,D(64),Y(2),I,J,K,K
1,Q,Q1,L1(2),L2(2),G,O
114 INTEGER Z(3,6),J1,J2,D3(32),L3(2,16),L4(2,16),A(2)
116 SHORT C(9,64),V(64),F(64),F1(32),Z1(6),V1,F2(32)
118 REAL T0,T1(2),T2,T3,T4,T5,T6,T7(2),T8(2),T9,U
120 D2=DATE @ T2=TIME
122 CLEAR
124 DISP USING 126
126 IMAGE 9X,"MODULE 'ACQUIR'",2/
128 DISP "TIME: ",HMS$(T2)
130 DISP "DATE: ",MDY$(D2+MDY("12/31/1981"))
132 DISP USING 134
134 IMAGE 3/,".....Wait"
136 WAIT 2000 @ ENABLE KBD 0
138 !
140 ! Read parameter file.
142 ASSIGN# 1 TO "PFILE"
144 CRT OFF
146 READ# 1 ; N,M,T,E(),X(),R(),S(),C(,),P$,H$
148 CLEAR
150 ASSIGN# 1 TO *
152 !
154 ! Initialize variables.
156 D1(1)=D2 @ D1(2)=D2
158 T1(1)=T2 @ T1(2)=T2
160 Y(1)=0 @ Y(2)=0
162 A(1)=0 @ A(2)=0
164 FOR I=1 TO 6
166 Z1(I)=0
168 FOR J=1 TO 3
170 Z(J,I)=0
172 NEXT J
174 NEXT I
175 FOR J=1 TO 32
176 F1(J)=0
177 NEXT J
178 ! Create output files.
180 ON ERROR GOTO 192
182 ASSIGN# 1 TO "FILE01"
184 OFF ERROR
186 PRINT# 1,1 ; P$[1,30],H$[1,256]
188 READ# 1,2 ; Y(1),D1(1),T1(1)
190 GOTO 203
192 OFF ERROR
194 CREATE "FILE01",102,586
196 ASSIGN# 1 TO "FILE01"
```

```
198 CRT OFF
200 PRINT# 1,1 ; P$[1,30],H$[1,256]
202 PRINT# 1,2 ; Y(1),D1(1),T1(1),D2,T2,1
203 IF N=1 THEN GOTO 230
204 ON ERROR GOTO 218
206 ASSIGN# 2 TO "FILE02"
208 OFF ERROR
210 PRINT# 2,1 ; P$[31,60],H$[257,512]
212 READ# 2,2 ; Y(2),D1(2),T1(2)
214 OFF ERROR
216 GOTO 234
218 OFF ERROR
220 CREATE "FILE02",102,586
222 ASSIGN# 2 TO "FILE02"
224 CRT OFF
226 PRINT# 2,1 ; P$[31,60],H$[257,512]
228 PRINT# 2,2 ; Y(2),D1(2),T1(2),D2,T2,2
230 !
232 ! Key assignments.
234 ENABLE KBD 32
236 ON KEY# 1,"MANU 1" GOTO 598
238 ON KEY# 2,"MANU 2" GOTO 606
240 ON KEY# 3,"AUTO 1" GOTO 694
242 ON KEY# 4,"AUTO 2" GOTO 698
244 ON KEY# 5,"DISP 1" GOTO 750
246 ON KEY# 6,"DISP 2" GOTO 754
248 ON KEY# 7,"HELP  " GOTO 836
250 ON KEY# 8,"EXIT  " GOTO 880
252 CLEAR @ KEY LABEL
254 DISP USING 256
256 IMAGE 3/,"   *****'ACQUIR' options*****",2/
258 DISP USING 260 ; Y(1),Y(2)
260 IMAGE "Proj #1: ",DDD," scan(s)"/"Proj #2: ",DDD, " scan(s)",2/
262 FOR I=1 TO 2
264 IF A(I)=0 THEN 270
266 DISP USING 268 ; I,HMS$(T7(I))
268 IMAGE "AUTO ",D," åctive at  ",K
270 NEXT I
272 GOTO 272
274 !
276 ! Call to 'MICROMUX'.
278 O=0
280 IOBUFFER B$ ! Usable 196
282 ASSERT 10;(2+4) ! DCD+DSR
284 ON TIMER# 3,30000 GOTO 336
286 STATUS 10,3 ; I ! Modem
288 IF BINAND(I,1)=0 THEN 286
290 WAIT 50 ! Wait for relay
292 SET TIMEOUT 10;8000
294 ON TIMEOUT 10 GOTO 342
296 ON EOT 10 GOTO 312
298 STATUS 10,9 ; I
```

```
300 CONTROL 10,9 ; 133 ! Flush
302 OUTPUT 10 USING "#,Z,A" ; M,"?"
304 ASSERT 10;(1+2+4) ! CTS on
306 ON TIMER# 3,8000 GOTO 342
308 TRANSFER 10 TO B$ INTR ; COUNT T*49
310 GOTO 310 ! Wait
312 OFF TIMER# 3
314 ASSERT 10;4 ! DCD&CTS off
316 TO=TIME
318 DO=DATE
320 FOR I=T TO 1 STEP -1
322 K=16*(I-1)
324 FOR J=1 TO 16
326 ENTER B$ USING "#,3Z" ; D(J+K)
328 NEXT J
330 ENTER B$ USING "#,X"
332 NEXT I
334 GOTO 354
336 O=1
338 BEEP @ PRINT "*****ERROR - MICROMUX IS BUSY"
340 GOTO 346
342 O=1
344 BEEP @ PRINT "*****ERROR - NO DATA RECIEVED"
346 OFF TIMER# 3
348 OFF TIMEOUT 10
350 OFF EOT 10 @ HALT 10
352 ASSERT 10;0 ! All off
354 RETURN
356 !
358 ! Change to engin. unit.
360 L1(1)=-1 @ L1(2)=-1
362 L2(1)=-1 @ L2(2)=-1
364 FOR I=1 TO 16
366 L3(1,I)=0 @ L3(2,I)=0
368 L4(1,I)=0 @ L4(2,I)=0
370 NEXT I
372 J=E(G)
374 ON ERROR GOTO 394
376 V(J)=C(1,J)*D(J)+C(2,J)
378 F(J)=V(J)
380 V1=V(J)
382 K=IP(J/17)+1
384 IF G=2 THEN K=IP((J-32)/17)+1
386 K1=J-16*K-32*G+48
388 IF F(J)<C(8,J) THEN L3(K,K1)=1
390 IF F(J)>C(9,J) THEN L4(K,K1)=1
392 GOTO 398
394 PRINT USING 396 ; G
396 IMAGE "*****ERROR - CONVERSION OF DATA",/,"EXCITATION CHANNEL, PRO
JECT",3D
398 FOR I=32*(G-1)+1 TO 32*G
400 ON ERROR GOTO 478
```

```
402 IF I=J THEN 494
404 V(I)=C(1,I)*D(I)+C(2,I)
406 Q=IP(I/17)+1
408 IF G=2 THEN Q=IP((I-32)/17)+1
410 Q1=I-16*Q-32*G+48
412 IF R(I)<11 THEN 428
414 J1=IP(R(I)/10)
416 J2=RMD(R(I),10)
418 IF J1>3 THEN 464
420 IF J2<1 THEN 464
422 IF J2>6 THEN 464
424 Z(J1,J2)=I
426 ON X(G) GOTO 438,452
428 ON R(I)+1 GOTO 430,434,438,442,448,452,460,464,464,464,464
430 F(I)=V(I)
432 GOTO 470
434 F(I)=C(3,I)*V(I)-C(4,I)
436 GOTO 470
438 F(I)=C(4,I)*V(I)/(C(6,I)*V1)-C(5,I)/C(6,I)
440 GOTO 470
442 U=LOG((V1/V(I)-1)*C(4,I))
444 F(I)=1/(C(3,I)*U+C(4,I))-C(5,I)
446 GOTO 470
448 F(I)=C(4,I)*V(I)/(C(6,I)*V1)+C(5,I)/C(6,I)+C(3,I)
450 GOTO 470
452 F(I)=C(3,I)*D(I)*V1/5.0075+C(4,I)
454 ! NEW EXTENSOMETER
456 ! REDUCTION FORMULA
458 GOTO 470
460 F(I)=4*V(I)/(V1*C(4,I))-C(5,I)
462 GOTO 470
464 V(I)=0
466 F(I)=0
468 GOTO 494
470 IF R(I)>10 THEN 494
472 IF F(I)<C(8,I) THEN L3(Q,Q1)=1
474 IF F(I)>C(9,I) THEN L4(Q,Q1)=1
476 GOTO 494
478 IF O=1 THEN 484
480 O=1 @ BEEP @ PRINT USING 482 ; Y(G)+1
482 IMAGE "*****ERROR - CONVERSION OF SCAN:",/,DDD,"    CHANNEL(S)"
484 PRINT USING 486 ; I
486 IMAGE 15X,DDD
488 V(I)=0 @ F(I)=0
490 L3(Q,Q1)=1
492 L4(Q,Q1)=1
494 NEXT I
496 FOR J1=1 TO 3
498 ON ERROR GOTO 502
500 Z1(1)=F(Z(J1,1))
502 FOR J2=2 TO 5
504 ON ERROR GOTO 508
```

```
506 Z1(J2)=Z1(1)-F(Z(J1,J2))
508 NEXT J2
510 FOR J2=1 TO 5
512 ON ERROR GOTO 532
514 I=Z(J1,J2)
516 Z(J1,J2)=0
518 F(I)=Z1(J2)
520 Z1(J2)=0
522 Q=IP(I/17)+1
524 IF G=2 THEN Q=IP((I-32)/17)+1
526 Q1=I-16*Q-32*G+48
528 IF F(I)<C(8,I) THEN L3(Q,Q1)=1
530 IF F(I)>C(9,I) THEN L4(Q,Q1)=1
532 NEXT J2
534 NEXT J1
536 OFF ERROR
538 FOR I=1 TO 2
540 FOR J=1 TO 15
542 IF L3(I,J)=0 THEN L1(I)=L1(I)-2^(J-1)
544 IF L4(I,J)=0 THEN L2(I)=L2(I)-2^(J-1)
546 NEXT J
548 IF L3(I,16)=0 THEN L1(I)=BINEOR(L1(I),-32768)
550 IF L4(I,16)=0 THEN L2(I)=BINEOR(L2(I),-32768)
552 NEXT I
554 RETURN
556 !
558 ! Write record on files.
560 J1=32*(G-1)+1
562 J2=32*G
564 K=0
566 FOR I=J1 TO J2
568 K=K+1
569 F2(I)=F(I)-F1(K)
570 F1(K)=F(I)
572 D3(K)=D(I)
574 NEXT I
576 IF Y(G)>1 THEN 580
578 D1(G)=D0 @ T1(G)=T0
580 CRT OFF
582 PRINT# G,2 ; Y(G),D1(G),T1(G),D0,T0,G
584 PRINT# G,Y(G)+2 ; T0,D0,F1(),D3(),L1(),L2()
586 CRT ON
588 RETURN
590 !
592 ! Manu, Auto project # 1,2.
594 ON TIMER# 1,T8(1)*60000 GOTO 594
596 T7(1)=RMD(TIME+T8(1)*60,86400)
598 G=1
600 GOTO 608
602 ON TIMER# 2,T8(2)*60000 GOTO 602
604 T7(2)=RMD(TIME+T8(2)*60,86400)
606 G=2
```

```
608 IF Y(G)<100 THEN 618
610 BEEP
612 PRINT USING 614 ; G
614 IMAGE "*****ERROR - MAX # OF SCANS",/,"FOR PROJECT # ",D," HAS BEE
N",/,"REACHED."
616 GOTO 234
618 CLEAR @ ENABLE KBD 0
620 GOSUB 278
622 IF O#0 THEN 234
624 GOSUB 360
626 Y(G)=Y(G)+1
628 GOSUB 560
630 CRT OFF
632 PRINT USING 634 ; G
634 IMAGE ,3/,5X,"*****PROJECT  ",D," *****"
636 PRINT USING 638 ; Y(G)
638 IMAGE "SCAN # ",DDD
640 PRINT "TIME: ",HMS$(TO)
642 PRINT "DATE: ",MDY$(DO+MDY("12/31/1981"))
644 PRINT USING 645
645 IMAGE 2/,"CHANNEL    ENGR.     NET    uMUXNAME       UNITS     CHA
NGE DATA"
646 PRINT "--------------------------------"
647 FOR I=21 TO 30
648 F(I)=F(I)*1000000 @ F2(I)=F2(I)*1000000
649 NEXT I
650 FOR I=32*(G-1)+1 TO 32*G
652 IF S(I)=0 THEN 682
654 Q=IP(I/17)+1
656 IF G=2 THEN Q=IP((I-32)/17)+1
658 Q1=I-16*Q-32*G+47
660 K=8*(I-1)+1
662 K1=K+7
664 J=0
666 IF BIT(L1(Q),Q1) THEN J=J+1
668 IF BIT(L2(Q),Q1) THEN J=J+2
670 IF J=0 THEN A$="   "
672 IF J=1 THEN A$="LOW"
674 IF J=2 THEN A$="HIG"
676 IF J=3 THEN A$="ERR"
678 PRINT USING 680 ; H$[K,K1],F(I),F2(I),D(I)
680 IMAGE 8A,X,MDDDDD.DDD, X,MDDD.DD,X,3Z
682 NEXT I
684 PRINT USING 686
686 IMAGE 5/
688 GOTO 234
690 !
692 ! Auto, project # 1,2.
694 G=1
696 GOTO 700
698 G=2
700 IF A(G)=0 THEN 708
```

```
702 A(G)=0
704 OFF TIMER# G
706 GOTO 234
708 A(G)=1
710 CLEAR
712 ENABLE KBD 1
714 ON ERROR GOTO 710
716 LINPUT "Input time interval between scans (MIN=3 min,MAX=1440 min)
",A$
718 T3=VAL(A$)
720 IF T3<3 THEN 710
722 IF T3>1440 THEN 710
724 LINPUT "Input time elapse befor start   (MIN=0 min,MAX=1440 min)",
A$
726 T9=VAL(A$)
728 IF T9<0 THEN 724
730 IF T9>1440 THEN 724
732 IF T9=0 THEN T9=.0005
734 OFF ERROR
736 T8(G)=T3
738 IF G=1 THEN ON TIMER# G,T9*60000 GOTO 594
740 IF G=2 THEN ON TIMER# G,T9*60000 GOTO 602
742 T7(G)=RMD(TIME+T9*60,86400)
744 GOTO 234
746 !
748 ! Display of data.
750 G=1
752 GOTO 756
754 G=2
756 CLEAR @ ENABLE KBD 0
758 GOSUB 278
760 IF O#0 THEN 234
762 GOSUB 360
764 J1=32*(G-1)+1
766 J2=32*G
768 I=J1-1
770 ENABLE KBD 1
772 DISP "Input channel # to be displayed"
774 DISP USING 776 ; J1,J2
776 IMAGE 2/,2X,DD,"-",DD,3X,"= channel for display",/,"'END LINE'= ne
xt channel"
778 DISP USING 780
780 IMAGE 4X,"0",5X,"= exit display",3/
782 ON ERROR GOTO 800
784 LINPUT "Channel #:",A$
786 J=VAL(A$)
788 OFF ERROR
790 IF J=0 THEN 234
792 IF J<J1 THEN 782
794 IF J>J2 THEN 782
796 I=J
798 GOTO 806
```

```
800 OFF ERROR
802 I=I+1
804 IF I>J2 THEN I=J1
806 Q=IP(I/17)+1
808 IF G=2 THEN Q=IP((I-32)/17)+1
810 Q1=I-16*Q-32*G+47
812 K=8*(I-1)+1
814 K1=K+7
816 CLEAR
818 DISP USING 820 ; I,H$[K,K1],F(I)
820 IMAGE "Channel #",6X,"= ",DD,/,"Title",10X,"= ",K,/,"Channel value
",2X,"= ",MD.DDDe
822 DISP USING 824 ; D(I),V(I)
824 IMAGE "Raw data",7X,"= ",3Z,/,"Voltage",8X,"= ",MDD.DDD
826 DISP USING 828 ; BIT(L1(Q),Q1),BIT(L2(Q),Q1)
828 IMAGE "Low",12X,"= ",D,/,"High",11X,"= ",D,8/
830 GOTO 782
832 !
834 ! Help.
836 ENABLE KBD 1 @ CLEAR
838 DISP USING 840
840 IMAGE 9X,"*****HELP*****",2/
842 DISP "The keys operate as follows:"
844 DISP USING 846
846 IMAGE 1/,"K1= 'MANUAL' acquisition proj.1",/,"K2= 'MANUAL' acquisi
tion proj.2"
848 DISP USING 850
850 IMAGE "K3= 'AUTO' acquisition proj.1",/,"K4= 'AUTO' acquisition pr
oj.2"
852 DISP USING 854
854 IMAGE "K5= 'DISPLAY' data for proj.1",/,"K6= 'DISPLAY' data for pr
oj.2"
856 DISP USING 858
858 IMAGE "K7= 'HELP'",/,"K8= 'EXIT' module 'ACQUIR'",3/
860 LINPUT "Hit 'END LINE' for more help",A$
862 CLEAR
864 DISP USING 866
866 IMAGE 2/,"Note: the max # of scans that can be stored for each pro
j. is      100."
868 DISP USING 870
870 IMAGE "In 'DISPLAY' mode, the data",/,"taken will not be stored in
",/,"files.",7/
872 LINPUT "Hit 'END LINE' to return",A$
874 GOTO 234
876 !
878 ! Exit.
880 CLEAR @ ENABLE KBD 0
882 OFF TIMER# 1 @ OFF TIMER# 2
884 ASSIGN# 1 TO *
886 ASSIGN# 2 TO *
888 ENABLE KBD 33 @ CRT ON
890 CHAIN "CONTRL"
892 END
```

MODULE 'INPUT'

```
100 ! This is module 'INPUT'.
102 ! It is the input segment
104 ! of data acquisition
106 ! program 'MDAP85'.
108 OPTION BASE 1
110 DIM P$[64],H$[512],Z$[512]
112 INTEGER M,T,N,E(2),X(2),R(64),S(64)
114 SHORT C(9,64)
116 CLEAR
118 DISP USING 120
120 IMAGE 9X,"MODULE 'INPUT'",3/,".....Wait"
124 ENABLE KBD 0
126 !
128 ! Read file.
130 ON ERROR GOTO 144
132 ASSIGN# 1 TO "PFILE"
134 CRT OFF
136 READ# 1 ; N,M,T,E(),X(),R(),S(),C(,),P$,H$
138 CLEAR
140 ASSIGN# 1 TO *
142 OFF ERROR @ GOTO 190
144 OFF ERROR
146 !
148 ! Initialize variables.
150 M=0
152 T=4
154 N=2
156 E(1)=0
158 E(2)=0
160 X(1)=0
162 X(2)=0
164 FOR I=1 TO 64
166 R(I)=0
168 S(I)=1
170 P$[I]="."
172 FOR J=1 TO 9
174 C(J,I)=0
176 NEXT J
178 NEXT I
180 FOR I=1 TO 512
182 H$[I]="."
184 NEXT I
186 !
188 ! Key assignments.
190 ENABLE KBD 32
192 P0=0
194 ON KEY# 1,"GENERAL" GOTO 220
196 ON KEY# 2,"PROJECT" GOTO 278
198 ON KEY# 3,"CHANNEL" GOTO 354
200 ON KEY# 4,"PRINT 1" GOTO 470
202 ON KEY# 5,"PRINT 2" GOTO 474
204 ON KEY# 6,"HELP   " GOTO 530
```

```
206 ON KEY# 7,"EXIT    " GOTO 578
208 CLEAR @ KEY LABEL
210 DISP USING 212
212 IMAGE 3/,"   *****'INPUT' options*****"
214 GOTO 214
216 !
218 ! General inputs.
220 ENABLE KBD 33
222 ALPHA 1,1 @ CLEAR
224 L1=CURSROW
226 DISP USING 228
228 IMAGE 4X,"*****GENERAL INPUTS*****",3/
230 DISP USING 232 ; M
232 IMAGE "Address of 'MICROMUX'= ",DD
234 DISP USING 236 ; T
236 IMAGE "# of 'MICROMUX' trans= ",DD
238 DISP USING 240 ; N
240 IMAGE "# of projects        = ",DD,9/
242 IF P0#0 THEN 492
244 ALPHA L1+13,1
246 ON ERROR GOTO 250
248 LINPUT "Inputs ok(Y)?",A$
250 ALPHA L1,1 @ OFF CURSOR
252 AREAD Z$
254 ON ERROR GOTO 222
256 M=ABS(VAL(TRIM$(Z$[119,128])))
258 T=ABS(VAL(TRIM$(Z$[151,160])))
260 N=ABS(VAL(TRIM$(Z$[183,192])))
262 IF T<1 THEN 222
264 IF T>4 THEN 222
266 IF N<1 THEN 222
268 IF N>2 THEN 222
270 OFF ERROR
272 GOTO 190
274 !
276 ! Project inputs.
278 ENABLE KBD 33
280 FOR I=1 TO N
282 ALPHA 1,1 @ CLEAR
284 L1=CURSROW
286 DISP USING 288 ; I
288 IMAGE 3X,"*****PROJECT INPUTS(",Z,")*****",3/
290 ON ERROR GOTO 282
292 LINPUT "Project #:",K$
294 N1=VAL(K$)
296 IF N1<1 THEN 282
298 IF N1>2 THEN 282
300 J1=1+30*(N1-1)
302 J2=J1+29
304 DISP USING 306 ; P$[J1,J2]
306 IMAGE "Project ID (max is 30 chars.)=",/,K,2/
308 DISP USING 310 ; E(N1)
```

```
310 IMAGE "Channel # for excit  = ",DD
312 DISP USING 314 ; X(N1)
314 IMAGE "'EXTENSOMETER' type  = ",DD,5/
316 IF P0#0 THEN 512
318 ALPHA L1+13,1
320 ON ERROR GOTO 324
322 LINPUT "Inputs ok(Y)?",A$
323 IF A$="Y" THEN 190
324 ALPHA L1,1 @ OFF CURSOR
326 AREAD Z$
328 ON ERROR GOTO 282
330 P$[J1,J2]=Z$[193,222]
332 E(N1)=ABS(VAL(TRIM$(Z$[279,288])))
334 X(N1)=ABS(VAL(TRIM$(Z$[311,320])))
336 IF E(N1)<1+32*(N1-1) THEN 282
338 IF E(N1)>32*N1 THEN 282
340 IF X(N1)<1 THEN 282
342 IF X(N1)>2 THEN 282
344 OFF ERROR
346 NEXT I
348 GOTO 190
350 !
352 ! Channel inputs.
354 ENABLE KBD 33
356 CLEAR
358 DISP USING 360
360 IMAGE 4X,"*****CHANNEL INPUTS*****"
362 DISP USING 364
364 IMAGE 3/,"Input channel # for each ",/,"channel."
366 DISP USING 368
368 IMAGE /,"Note: a '0' for channel # will",/,"exit this routine",2/
370 ON ERROR GOTO 372
372 LINPUT "Channel #:",K$
374 N1=VAL(K$)
376 IF N1=0 THEN 462
378 IF N1<0 THEN 372
380 IF N1>64 THEN 372
382 ALPHA 1,1 @ CLEAR
384 L1=CURSROW
386 OFF ERROR
388 DISP USING 390 ; N1
390 IMAGE 7X,"***Channel ",2D,"  ***",
392 J1=8*(N1-1)+1
394 J2=J1+7
396 DISP USING 398 ; H$[J1,J2]
398 IMAGE "Title (8 char) =  ",K
400 DISP USING 402 ; R(N1)
402 IMAGE "Reduction type = ",DD
404 DISP USING 406 ; S(N1)
406 IMAGE "Printing flag  = ",DD
408 DISP USING 410 ; C(1,N1),C(2,N1),C(3,N1)
410 IMAGE "C0",13X,"= ",MD.DDDDE,/,"C1",13X,"= ",MD.DDDDE,/,"C2",13X,"
```

```
= ",MD.DDDDE
412 DISP USING 414 ; C(4,N1),C(5,N1),C(6,N1)
414 IMAGE "C3",13X,"= ",MD.DDDDE,/,"C4",13X,"= ",MD.DDDDE,/,"C5",13X,"
= ",MD.DDDDE
416 DISP USING 418 ; C(7,N1)
418 IMAGE "C6",13X,"= ",MD.DDDDE
420 DISP USING 422 ; C(8,N1),C(9,N1)
422 IMAGE "Low limit",6X,"= ",MD.DDDDE,/,"High limit",5X,"= ",MD.DDDDE
,2/
424 IF P0#0 THEN 520
426 ALPHA L1+13,1
428 ON ERROR GOTO 432
430 LINPUT "Inputs ok(Y)?",A$
432 ALPHA L1,1 @ OFF CURSOR
434 AREAD Z$
436 ON ERROR GOTO 382
438 H$[J1,J2]=TRIM$(Z$[49,64])
440 R(N1)=ABS(VAL(TRIM$(Z$[81,96])))
442 S(N1)=ABS(VAL(TRIM$(Z$[113,128])))
444 IF R(N1)>36 THEN 382
446 IF S(N1)>1 THEN 382
448 FOR I=1 TO 9
450 J1=32*(I-1)+145
452 J2=J1+15
454 C(I,N1)=VAL(TRIM$(Z$[J1,J2]))
456 NEXT I
458 OFF ERROR
460 CLEAR @ GOTO 370
462 OFF ERROR
464 GOTO 190
466 !
468 ! Print informations.
470 P0=1
472 GOTO 476
474 P0=2
476 CLEAR @ ENABLE KBD 0
478 CRT OFF @ CRT IS 2
480 DISP "-------INPUT INFORMATION-------"
482 DISP "TIME: ",HMS$(TIME)
484 DISP "DATE: ",MDY$(DATE+MDY("12/31/1981"))
486 DISP "-------------------------------"
488 DISP " "
490 GOTO 226
492 DISP USING 494
494 IMAGE 4X,"*****PROJECT INPUTS*****",3/
496 N1=P0
498 DISP USING 500 ; N1
500 IMAGE "Project # =",3D
502 J1=1+30*(N1-1)
504 J2=J1+29
506 DISP USING 508 ; P$[J1,J2]
508 IMAGE "Project ID=",/,K,2/
```

```
510 GOTO 308
512 DISP USING 514
514 IMAGE 4X,"*****CHANNEL INPUTS*****",3/
516 FOR N1=1+32*(PO-1) TO 32*PO
518 GOTO 388
520 NEXT N1
522 CRT IS 1 @ CRT ON
524 GOTO 190
526 !
528 ! Help.
530 ENABLE KBD 1 @ CLEAR
532 DISP USING 534
534 IMAGE 9X,"*****HELP*****",2/
536 DISP "The keys operate as follows:"
538 DISP USING 540
540 IMAGE 1/,"K1= 'GENERAL' inputs",/,"K2= 'PROJECT' inputs",/,"K3= 'C
HANNEL' inputs"
542 DISP USING 544
544 IMAGE "K4= 'PRINT' inputs for proj. 1",/,"K5= 'PRINT' inputs for p
roj. 2"
546 DISP USING 548
548 IMAGE "K6= 'HELP'",/,"K7= 'EXIT' module 'INPUT'",4/
550 LINPUT "Hit 'END LINE' for more help",A$
552 CLEAR
554 DISP USING 556
556 IMAGE 2/, "Note: in input, when the",/,"current values of paramete
rs"
558 DISP USING 560
560 IMAGE "have been presented on CRT,",/,"roll the CURSOR to the appr
opr-"
562 DISP USING 564
564 IMAGE "iate locations for changing the",/,"inputs.",/,"After all c
hanges are done,"
566 DISP USING 568
568 IMAGE "hit 'END LINE' to exit.",5/
570 LINPUT "Hit 'END LINE' to return",A$
572 GOTO 190
574 !
576 ! Exit.
578 CLEAR
580 ON ERROR GOTO 584
582 CREATE "PFILE",1,6800
584 ON ERROR GOTO 600
586 ENABLE KBD 0
588 ASSIGN# 1 TO "PFILE"
590 CRT OFF
592 PRINT# 1 ; N,M,T,E(),X(),R(),S(),C(,),P$,H$
594 CRT ON
596 ASSIGN# 1 TO *
598 ENABLE KBD 255
600 ON ERROR GOTO 604
602 CHAIN "CONTRL"
604 END
```

MODULE 'STORE'

```
100 ! This is module 'STORE'.
102 ! It is the data storage
104 ! segment of prog. 'MDAP85'.
108 OPTION BASE 1
110 DIM P$[30],H$[256],Z$[6]
112 INTEGER L1(100),L2(100),L3(100),L4(100),D(100),D1,D2,Y,G,J,K,D3(10
0,32),D4(32),Y1
114 SHORT F(100,32),F1(32)
116 REAL T(100),T1,T2
118 CLEAR
120 DISP "          MODULE 'STORE'"
122 WAIT 3000
124 ! Key assignments.
126 ENABLE KBD 32
128 ON KEY# 1,"STORE 1" GOTO 148
130 ON KEY# 2,"STORE 2" GOTO 158
132 ON KEY# 3,"HELP   " GOTO 334
134 ON KEY# 4,"EXIT   " GOTO 366
136 CLEAR @ KEY LABEL
138 DISP USING 140
140 IMAGE 3/,"    *****'STORE' options*****"
142 GOTO 142
144 !
146 ! Store 'FILE01','FILE02'.
148 G=1
150 CLEAR @ ENABLE KBD 0
152 ON ERROR GOTO 192
154 ASSIGN# 1 TO "FILE01"
156 GOTO 166
158 G=2
160 CLEAR @ ENABLE KBD 0
162 ON ERROR GOTO 192
164 ASSIGN# 1 TO "FILE02"
166 OFF ERROR
168 DISP USING 170
170 IMAGE 3/,".....Wait"
172 WAIT 3000
174 CLEAR @ CRT OFF
176 READ# 1,1 ; P$,H$
178 READ# 1,2 ; Y,D1,T1,D2,T2
180 IF Y>0 THEN 200
182 BEEP
184 IF G=1 THEN PRINT "*****ERROR - NO DATA ('FILE01')"
186 IF G=2 THEN PRINT "*****ERROR - NO DATA ('FILE02')"
188 ASSIGN# 1 TO *
190 GOTO 126
192 OFF ERROR @ BEEP
194 IF G=1 THEN PRINT "*****ERROR - ASSIGN 'FILE01'"
196 IF G=2 THEN PRINT "*****ERROR - ASSIGN 'FILE02'"
198 GOTO 126
200 Y1=0 @ ON ERROR GOTO 218
202 FOR J=1 TO Y
```

```
204 READ# 1,J+2 ; T(J),D(J),F1(),D4(),L1(J),L2(J),L3(J),L4(J)
206 Y1=Y1+1
208 FOR K=1 TO 32
210 F(J,K)=F1(K)
212 D3(J,K)=D4(K)
214 NEXT K
216 NEXT J
218 OFF ERROR
220 ASSIGN# 1 TO *
222 ENABLE KBD 1
224 CLEAR
226 LINPUT "Insert the mass storage tape     and hit 'END LINE'",A$
228 CLEAR @ CRT IS 2
230 ON ERROR GOTO 256
232 BEEP
234 DISP USING 236
236 IMAGE 2/,"---------TAPE DIRECTORY--------"
238 CAT
240 CRT IS 1
242 ON ERROR GOTO 244
244 LINPUT "Input new file name(1-6 char):",A$
246 Z$=TRIM$(A$)
248 ON ERROR GOTO 256
250 CREATE Z$,Y1+2,586
252 OFF ERROR
254 GOTO 276
256 OFF ERROR @ CRT IS 1 @ CRT ON
258 IF ERRN=60 THEN BEEP @ PRINT "*****ERROR - TAPE IS WRITE      PROT
ECTED"
260 IF ERRN=60 THEN 224
262 IF ERRN=62 THEN 224
264 IF ERRN=73 THEN ERASETAPE
266 IF ERRN=73 THEN 228
268 IF ERRN=65 THEN BEEP @ PRINT "*****ERROR - TAPE IS FULL"
270 IF ERRN=63 THEN BEEP @ PRINT "*****ERROR - DUPLICATED FILE     NAM
E"
272 IF ERRN=65 THEN 224
274 GOTO 242
276 CLEAR @ ENABLE KBD 0
278 ASSIGN# 1 TO Z$
280 CRT OFF
282 PRINT# 1,1 ; P$,H$
284 PRINT# 1,2 ; Y1,D1,T1,D2,T2,G
286 FOR J=1 TO Y1
288 FOR K=1 TO 32
290 F1(K)=F(J,K)
292 D4(K)=D3(J,K)
294 NEXT K
296 PRINT# 1,J+2 ; T(J),D(J),F1(),D4(),L1(J),L2(J),L3(J),L4(J)
298 NEXT J
300 ASSIGN# 1 TO *
302 CLEAR @ ENABLE KBD 1
```

```
303 CAT
304 LINPUT "Put the program tape back and   hit 'END LINE'",A$
306 ON ERROR GOTO 308
308 LINPUT "Purge the original file(Y/N)?",A$
310 OFF ERROR
312 IF A$="N" THEN 126
314 IF A$#"Y" THEN 306
316 ON ERROR GOTO 326
318 IF G=1 THEN ASSIGN# 1 TO "FILE01"
320 IF G=2 THEN ASSIGN# 1 TO "FILE02"
322 PRINT# 1,2 ; 0,D1,T1,D2,T2,G
324 ASSIGN# 1 TO *
326 OFF ERROR
328 GOTO 126
330 !
332 ! Help.
334 ENABLE KBD 1 @ CLEAR
336 DISP USING 338
338 IMAGE 9X,"*****HELP*****",2/
340 DISP "The keys operate as follows:"
342 DISP USING 344
344 IMAGE /,"K1= 'STORE' data for project 1",/,"K2= 'STORE' data for p
roject 2"
346 DISP USING 348
348 IMAGE "K3= 'HELP'",/,"K4= 'EXIT' module 'STORE'",2/
350 DISP USING 352
352 IMAGE "Note: the name inputted for the",/,"file to be stored shoul
d be"
354 DISP USING 356
356 IMAGE "different from those already in",/,"the tape directory.",2/

358 LINPUT "Hit 'END LINE' to return",A$
360 GOTO 126
362 !
364 ! Exit.
366 ENABLE KBD 33
368 CHAIN "CONTRL"
370 END
```

MODULE 'PRINT'

```
100 ! This is module 'PRINT'.
102 ! It is the print segment
104 ! of the data acquisition
106 ! program 'MDAP85'.
108 OPTION BASE 1
110 DIM P$[30],H$[256],A$[30],Z$[6]
112 INTEGER L1(2),L2(2),D0,D1,D2,D3,D4,G,G1,I,I1,J,J1,J2,K,K1,Q,Q1,O,D
(32)
114 SHORT F(32),F1(5,32),U
116 REAL T0,T1,T2,T3,T4
118 CLEAR @ DISP "         MODULE 'PRINT'"
120 WAIT 3000
122 !
124 ! Key assignments.
126 ENABLE KBD 32
128 G1=0
130 ON KEY# 1,"PRINT 1" GOTO 192
132 ON KEY# 2,"PRINT 2" GOTO 204
134 ON KEY# 3,"STORED " GOTO 152
136 ON KEY# 4,"HELP   " GOTO 440
138 ON KEY# 5,"EXIT   " GOTO 464
140 CLEAR @ KEY LABEL
142 DISP USING 144
144 IMAGE 3/,"   *****'PRINT' options*****"
146 GOTO 146
148 !
150 ! Print stored file.
152 ENABLE KBD 1
154 CLEAR
156 LINPUT "Insert the tape containing the  stored data and hit 'END L
INE'",A$
158 CLEAR @ CRT IS 2
160 ON ERROR GOTO 180
162 DISP USING 164
164 IMAGE 2/,"---------TAPE DIRECTORY--------"
166 CAT @ CRT IS 1
168 ON ERROR GOTO 170
170 LINPUT "Input file name(1-6 char):",A$
172 Z$=TRIM$(A$)
174 ON ERROR GOTO 180
176 ASSIGN# 1 TO Z$
178 GOTO 224
180 OFF ERROR @ CRT IS 1
182 IF ERRN=67 THEN BEEP @ PRINT "*****ERROR - NO SUCH FILE"
184 IF ERRN=67 THEN GOTO 430
186 GOTO 154
188 !
190 ! Project # 1,2.
192 G1=1
194 ENABLE KBD 0
196 CLEAR
198 ON ERROR GOTO 216
```

```
200 ASSIGN# 1 TO "FILE01"
202 GOTO 224
204 G1=2
206 ENABLE KBD 0
208 CLEAR
210 ON ERROR GOTO 216
212 ASSIGN# 1 TO "FILE02"
214 GOTO 224
216 OFF ERROR @ BEEP
218 IF G1=1 THEN PRINT "*****ERROR - ASSIGN 'FILE01'"
220 IF G1=2 THEN PRINT "*****ERROR - ASSIGN 'FILE02'"
222 GOTO 126
224 OFF ERROR @ ENABLE KBD 0
226 CLEAR @ CRT OFF
228 READ# 1,1 ; P$,H$
230 READ# 1,2 ; Y,D1,T1,D2,T2,G
232 CRT ON
234 IF Y>0 THEN 240
236 BEEP @ PRINT "*****ERROR - NO DATA"
238 GOTO 426
240 DISP USING 242 ; G,P$
242 IMAGE 5X,"*****PROJECT  ",D," *****",/,"Project ID:",/,K,2/
244 DISP USING 246 ; Y
246 IMAGE "# of scans: ",DDD
248 DISP USING 250 ; HMS$(T1),MDY$(D1+MDY("12/31/1981"))
250 IMAGE "Starting time and date:",/,K,10X,K
252 DISP USING 254 ; HMS$(T2),MDY$(D2+MDY("12/31/1981"))
254 IMAGE "Ending time and date:",/,K,10X,K,2/
256 ON ERROR GOTO 224
258 ENABLE KBD 1
260 DISP "Enter time interval for printingthe data:"
262 LINPUT "mm/dd/yyyy hh:mm:ss    (start)",A$
264 A$=TRIM$(A$)
266 D3=MDY(A$[1,10])-MDY("12/31/1981")
268 T3=HMS(TRIM$(A$[12]))
270 LINPUT "mm/dd/yyyy hh:mm:ss    (end)",A$
272 A$=TRIM$(A$)
274 D4=MDY(A$[1,10])-MDY("12/31/1981")
276 T4=HMS(TRIM$(A$[12]))
278 OFF ERROR
280 O=0
282 IF D3>D2 THEN O=1
284 IF D4<D1 THEN O=1
286 IF D3>D4 THEN O=1
288 IF O=0 THEN 298
290 BEEP @ PRINT "*****ERROR - WRONG INTERVAL"
292 GOTO 426
294 !
296 ! Print the data.
298 CRT OFF
300 PRINT USING 302
302 IMAGE 3/, "---------DATA PRINTOUT---------"
```

```
304 PRINT USING 306 ; G
306 IMAGE "PROJECT  #:",DD
308 PRINT USING 310 ; P$
310 IMAGE "PROJECT ID:",/,K,2/
312 PRINT USING 314 ; Y
314 IMAGE "TOTAL # OF SCANS: ",DDD
316 PRINT USING 318 ; HMS$(T1),MDY$(D1+MDY("12/31/1981"))
318 IMAGE "STARTING TIME AND DATE:",/,K,10X,K
320 PRINT USING 322 ; HMS$(T2),MDY$(D2+MDY("12/31/1981"))
322 IMAGE "ENDING TIME AND DATE:",/,K,10X,K
324 PRINT "--------------------------------"
326 O=0
328 FOR J=1 TO Y
330 CRT OFF
332 READ# 1,J+2 ; T0,D0,F(),D(),L1(),L2()
334 J1=RMD(J,5)+1
336 FOR I=1 TO 32
338 F1(J1,I)=F(I)
340 NEXT I
342 IF D0<D3 THEN 418
344 IF D0>D4 THEN 420
346 IF D0#D3 THEN 350
348 IF T0<T3 THEN 418
350 IF D0#D4 THEN 354
352 IF T0>T4 THEN 420
354 O=O+1
356 PRINT USING 358 ; J
358 IMAGE 4/,7X,"******SCAN ",DDD," ******",2/
360 PRINT "TIME: ",HMS$(T0)
362 PRINT "DATE: ",MDY$(D0+MDY("12/31/1981"))
364 PRINT USING 366
366 IMAGE /,"CH CHANNEL    CHANNEL   RAW",/," #   NAME      VALUE   DA
TA CON"
368 PRINT "--------------------------------"
370 FOR I=1 TO 32
372 Q=IP(I/17)+1
374 Q1=I-16*Q+15
376 K=8*(I-1)+1
378 K1=K+7
380 I1=0
382 IF BIT(L1(Q),Q1) THEN I1=I1+1
384 IF BIT(L2(Q),Q1) THEN I1=I1+2
386 IF I1=0 THEN A$="   "
388 IF I1=1 THEN A$="LOW"
390 IF I1=2 THEN A$="HIG"
392 IF I1=3 THEN A$="ERR"
394 IF I1=3 THEN 412
396 IF J<5 THEN 412
398 U=0
400 FOR J2=1 TO 5
402 IF J2=J1 THEN 406
404 U=U+F1(J2,I)/4
```

```
406 NEXT J2
408 IF U=0 THEN 412
410 IF ABS((F(I)-U)/U)>.05 THEN A$="CHG"
412 PRINT USING 414 ; I+32*G-32,H$[K,K1],F(I),D(I),A$
414 IMAGE DD,X,8A,X,MD.DDDe,2X,3Z,X,3A
416 NEXT I
418 NEXT J
420 IF O=O THEN BEEP @ PRINT "*****ERROR - NO DATA IN THE     GIVEN IN
TERVAL"
422 PRINT USING 424
424 IMAGE "-------------------------------",5/
426 ASSIGN# 1 TO *
428 IF G1#0 THEN 126
430 CLEAR @ ENABLE KBD 1
432 LINPUT "Put the program tape back and   hit 'END LINE'",A$
434 GOTO 126
436 !
438 ! Help.
440 CLEAR @ ENABLE KBD 1
442 DISP USING 444
444 IMAGE 9X,"*****HELP*****",2/
446 DISP "The keys operate as follows:"
448 DISP USING 450
450 IMAGE 1/,"K1= 'PRINT' project 1",/,"K2= 'PRINT' project 2"
452 DISP USING 454
454 IMAGE "K3= print a 'STORED' file",/,"K4= 'HELP'",/,"K5= 'EXIT' mod
ule 'PRINT'",6/
456 LINPUT "Hit 'END LINE' to return",A$
458 GOTO 126
460 !
462 ! Exit.
464 CLEAR @ ENABLE KBD 33
466 CHAIN "CONTRL"
468 END
```

MODULE 'PLOT'

```
100 ! This is module 'PLOT'.
102 ! It is the plot segment
104 ! of the data acquisition
106 ! program 'MDAP'.
108 OPTION BASE 1
110 DIM P$[30],H$[256],Z$[6],W$[30],P1$[1],A$[30]
112 INTEGER D(100),D1,D2,D3,D4,I,J,K,K1,G,G1,O,L,M,M1,N,Y
114 SHORT F(100,32),F1(32),Q
116 REAL T(100),T1,T2,T3,T4,X1,X2,X3,X4,Y1,Y2,Y3,Y4,E
118 CLEAR
120 DISP "          MODULE 'PLOT'"
122 WAIT 3000
124 !
126 ! Key assignments.
128 ENABLE KBD 32
130 G1=0 @ O=0
132 ON KEY# 1,"PLOT 1" GOTO 196
134 ON KEY# 2,"PLOT 2" GOTO 208
136 ON KEY# 3,"STORED" GOTO 154
138 ON KEY# 4,"HELP  " GOTO 610
140 ON KEY# 5,"EXIT  " GOTO 634
142 CLEAR @ KEY LABEL
144 DISP USING 146
146 IMAGE 3/,"    *****'PLOT' options*****"
148 GOTO 148
150 !
152 ! Plot stored file.
154 ENABLE KBD 1
156 CLEAR
158 LINPUT "Insert the tape containing the  stored data and hit 'END L
INE'",A$
160 CLEAR @ CRT IS 2
162 ON ERROR GOTO 184
164 DISP USING 166
166 IMAGE 2/,"---------TAPE DIRECTORY--------"
168 CAT @ CRT IS 1
170 ON ERROR GOTO 172
172 LINPUT "Input file name(1-6 char):",A$
174 Z$=TRIM$(A$)
176 ON ERROR GOTO 184
178 ENABLE KBD 0
180 ASSIGN# 1 TO Z$
182 GOTO 228
184 OFF ERROR @ CRT IS 1
186 IF ERRN#67 THEN 154
188 BEEP @ PRINT "*****ERROR - NO SUCH FILE"
190 O=1 @ GOTO 258
192 !
194 ! Project # 1,2.
196 G1=1
198 ENABLE KBD 0
200 CLEAR
```

```
202 ON ERROR GOTO 220
204 ASSIGN# 1 TO "FILE01"
206 GOTO 228
208 G1=2
210 ENABLE KBD 0
212 CLEAR
214 ON ERROR GOTO 220
216 ASSIGN# 1 TO "FILE02"
218 GOTO 228
220 OFF ERROR @ BEEP
222 IF G1=1 THEN PRINT "*****ERROR - ASSIGN 'FILE01'"
224 IF G1=2 THEN PRINT "*****ERROR - ASSIGN 'FILE02'"
226 GOTO 128
228 OFF ERROR
230 CLEAR @ CRT OFF
232 READ# 1,1 ; P$,H$
234 READ# 1,2 ; Y,D1,T1,D2,T2,G
236 IF Y>0 THEN 244
238 BEEP @ PRINT "*****ERROR - NO DATA"
240 ASSIGN# 1 TO *
242 GOTO 128
244 FOR I=1 TO Y
246 READ# 1,I+2 ; T(I),D(I),F1()
248 FOR J=1 TO 32
250 F(I,J)=F1(J)
252 NEXT J
254 NEXT I
256 ASSIGN# 1 TO *
258 ENABLE KBD 1
260 IF G1#0 THEN 276
262 !
264 ! Put program tape back.
266 CLEAR
268 LINPUT "Put the program tape back and   hit 'END LINE'",A$
270 IF O=1 THEN 128
272 !
274 ! Get infor. for plot.
276 CLEAR
278 DISP USING 280 ; G,P$
280 IMAGE 5X,"*****PROJECT  ",D," *****",/,"Project ID:",/,K,2/
282 DISP USING 284 ; Y
284 IMAGE "# of data points: ",DDD
286 DISP USING 288 ; MDY$(D1+MDY("12/31/1981")),HMS$(T1)
288 IMAGE "Starting date and time:",/,K,1X,K
290 DISP USING 292 ; MDY$(D2+MDY("12/31/1981")),HMS$(T2)
292 IMAGE "Ending date and time:",/,K,1X,K,2/
294 ON ERROR GOTO 276
296 DISP "Enter time interval for plottingthe data:"
298 LINPUT "mm/dd/yyyy hh:mm:ss    (start)",A$
300 A$=TRIM$(A$)
302 D3=MDY(A$[1,10])-MDY("12/31/1981")
304 T3=HMS(TRIM$(A$[12]))
```

```
306 LINPUT "mm/dd/yyyy hh:mm:ss    (end)",A$
308 A$=TRIM$(A$)
310 D4=MDY(A$[1,10])-MDY("12/31/1981")
312 T4=HMS(TRIM$(A$[12]))
314 OFF ERROR
316 O=0
318 IF D3>D2 THEN O=1
320 IF D4<D1 THEN O=1
322 IF D3>D4 THEN O=1
324 IF D3#D4 THEN 328
326 IF T3>=T4 THEN O=1
328 IF O=0 THEN 334
330 BEEP @ PRINT "*****ERROR - WRONG INTERVAL"
332 GOTO 276
334 CLEAR @ ON ERROR GOTO 334
336 DISP "YMIN,YMAX for plot";
338 INPUT Y3,Y4
340 LINPUT "Plot on CRT or PLOTTER(C/P)?",P1$
342 IF P1$#"P" THEN 348
344 LINPUT "Input LABEL for Y axis:",W$
346 LINPUT "Set plotter and hit 'END LINE'",A$
348 OFF ERROR
350 !
352 ! Draw axes and boundar.
354 X1=D3+T3/86400-365
356 X2=D4+T4/86400-365
358 Y1=Y3 @ Y2=Y4
360 IF P1$#"P" THEN 420
362 PLOTTER IS 705
364 LOCATE 14,100*RATIO-14,12,82
366 SCALE X1,X2,Y1,Y2
368 FXD 3,3 @ CSIZE 2,.5,0
370 LAXES
372 MOVE X1,Y2 @ DRAW X2,Y2
374 DRAW X2,Y1
376 CSIZE 3,.5,0
378 DEG @ LDIR 90
380 SETGU @ MOVE 2,40
382 LABEL W$
384 LDIR 0
386 MOVE 40,1
388 LABEL "TIME   (DAYS SINCE 12/31/1982)"
390 MOVE 14,95 @ CSIZE 4,.5,0
392 LABEL USING 394 ; P$
394 IMAGE K
396 MOVE 14,91 @ CSIZE 3,.5,0
398 LABEL USING 400 ; G+3
400 IMAGE "INSTALLATION: ",D
402 MOVE 14,89
404 A$[1,8]=HMS$(T3) @ A$[9,13]="     " @ A$[14,23]=MDY$(D3+MDY("12/31/
1981"))
406 LABEL USING 408 ; A$[1,23]
```

```
408 IMAGE "STARTING TIME AND DATE FOR PLOT:",5X,K
410 MOVE 14,87
412 A$[1,8]=HMS$(T4) @ A$[9,13]="     " @ A$[14,23]=MDY$(D4+MDY("12/31/
1981"))
414 LABEL USING 416 ; A$[1,23]
416 IMAGE "ENDING TIME AND DATE FOR PLOT:",7X,K
418 GOTO 442
420 PLOTTER IS 1
422 GCLEAR
424 LOCATE 35,100*RATIO-5,12,95
426 SCALE X1,X2,Y1,Y2
428 FXD 3,2
430 LAXES (X2-X1)/-24,(Y2-Y1)/24,X1,Y1,10,10
432 SETGU
434 MOVE 35,0
436 LABEL "DAYS SINCE 01/01/1982"
438 !
440 ! Get infor. for curve.
442 I=0
444 I=I+1
446 SETUU @ MOVE X1,Y1
448 IF I>6 THEN 590
450 CLEAR
452 LINPUT "Plot data(Y/N)?",A$
454 IF A$="N" THEN 590
456 IF A$#"Y" THEN 450
458 ON ERROR GOTO 460
460 LINPUT "Input channel # to be plotted:",A$
462 J=VAL(A$)
464 OFF ERROR
466 IF J<32*(G-1)+1 THEN 450
468 IF J>32*G THEN 450
470 IF G=2 THEN J=J-32
472 N=100 @ A$="Y"
474 IF P1$#"P" THEN 498
476 ON ERROR GOTO 478
478 LINPUT "Input # of points to bypass for identification:",A$
480 N=VAL(A$)
482 OFF ERROR
484 IF N<0 THEN 450
485 IF J<21 THEN 489
486 IF J>30 THEN 489
487 LINPUT "Input Modulas; Ex10^6, E=1 for Microstrain output",A$
488 E=VAL(A$)
489 LINPUT "Connect data points(Y/N)?",A$
490 IF A$="Y" THEN 498
491 IF A$="N" THEN 498
492 GOTO 486
494 !
496 ! Plot curve.
498 M=0 @ M1=0
500 IF P1$#"P" THEN GRAPHICS @ E=1000000
```

```
502 FOR L=1 TO Y-1
504 X3=D(L)+T(L)/86400-365
506 X4=D(L+1)+T(L+1)/86400-365
508 IF X3<X1 THEN 554
510 IF X4>X2 THEN 560
511 Y3=F(L,J) @ Y4=F(L+1,J)
512 IF J<21 THEN 515
513 IF J>30 THEN 515
514 Y3=Y3*E*1000000 @ Y4=Y4*E*1000000
515 M=M+1
516 IF M>N THEN M=0
518 IF M#0 THEN 550
520 SETUU @ MOVE X3,Y3 @ SETGU
522 ON I GOSUB 526,530,534,538,542,546
524 M1=1 @ GOTO 550
526 RPLOT .3,1,-1 @ RPLOT -.3,1,-1 @ RPLOT 0,0,2
528 RETURN
530 RPLOT .3,-1,-1 @ RPLOT -.3,-1,-1 @ RPLOT 0,0,2
532 RETURN
534 RPLOT -.5,.5,-1 @ RPLOT -.5,-.5,-1 @ RPLOT 0,0,2
536 RETURN
538 RPLOT .5,.5,-1 @ RPLOT .5,-.5,-1 @ RPLOT 0,0,2
540 RETURN
542 RPLOT -.5,0,1 @ RPLOT .5,0,-1 @ RPLOT 0,-.5,-2 @ RPLOT 0,.5,-1
544 RETURN
546 RPLOT -.5,-.5,1 @ RPLOT .5,.5,-1 @ RPLOT .5,-.5,-2 @ RPLOT -.5,.5,
-1
548 RETURN
550 SETUU @ IF A$#"Y" THEN 554
552 MOVE X3,Y3 @ DRAW X4,Y4
554 NEXT L
556 !
558 ! Write legend.
560 IF P1$#"P" THEN 444
562 IF M1=0 THEN 444
564 K=8*(J-1)+1
566 K1=K+7
568 Q=.5
570 IF I=1 THEN Q=0
572 IF I=2 THEN Q=1
574 SETGU @ MOVE 100*RATIO-26,80-I*2+Q
576 ON I GOSUB 526,530,534,538,542,546
578 MOVE 100*RATIO-24,80-I*2
580 LABEL USING 582 ; H$[K,K1]
582 IMAGE "=",K
584 GOTO 444
586 !
588 ! Plot more?
590 IF P1$#"C" THEN 594
592 GRAPHICS @ COPY
594 PEN 0
596 CLEAR
```

```
598 LINPUT "Plot more from this file(Y/N)?",A$
600 IF A$="Y" THEN 276
602 IF A$="N" THEN 128
604 GOTO 596
606 !
608 ! Help.
610 CLEAR @ ENABLE KBD 1
612 DISP USING 614
614 IMAGE 9X,"*****HELP*****",2/
616 DISP "The keys operate as follows:"
618 DISP USING 620
620 IMAGE 1/,"K1= 'PLOT' of project 1",/,"K2= 'PLOT' of project 2"
622 DISP USING 624
624 IMAGE "K3= plot of a 'STORED' file",/,"K4= 'HELP'",/,"K5= 'EXIT' m
odule 'PLOT'",6/
626 LINPUT "Hit 'END LINE' to return",A$
628 GOTO 128
630 !
632 ! Exit.
634 CLEAR @ ENABLE KBD 33
636 CHAIN "DATA"
638 END
```

MODULE 'NPLOT'

```
100 ! This is program 'NPLOT'.
120 ! It plots the files crea-
140 ! ted by program "ACQUIR".
160 OPTION BASE 1
180 DIM P$[30],H$[256],Z$[6],W$[30],A$[30]
200 INTEGER D(100),D1,D2,D3,D4,I,J,K,K1,G,G1,O,L,M,M1(6),N,C
220 SHORT F(100,32),F1(32),Q
240 REAL T(100),T1,T2,T3,T4,X1,X2,X3,X4,Y1,Y2,Y3,Y4,E
260 CLEAR
280 DISP "         PROGRAM 'PLOT'"
300 WAIT 3000
320 !
340 ! Initialize variables.
360 C=0
380 FOR I=1 TO 6
400 M1(I)=0
420 NEXT I
440 !
460 ! Key assignments.
480 ENABLE KBD 32
500 G1=0
520 ON KEY# 1,"PROJ 1" GOTO 1220
540 ON KEY# 2,"PROJ 2" GOTO 1340
560 ON KEY# 3,"STORED" GOTO 740
580 ON KEY# 4,"HELP  " GOTO 5240
600 ON KEY# 5,"EXIT  " GOTO 5480
620 CLEAR @ KEY LABEL
640 DISP USING 660
660 IMAGE 3/,"   *****'NPLOT' options*****"
680 GOTO 680
700 !
720 ! Plot stored file.
740 ENABLE KBD 1
760 CLEAR
780 LINPUT "Insert the tape containing the  stored data and hit 'END L
INE'",A$
800 CLEAR @ CRT IS 2
820 ON ERROR GOTO 1060
840 DISP USING 860
860 IMAGE 2/,"---------TAPE DIRECTORY--------"
880 CAT
900 CRT IS 1
920 ON ERROR GOTO 940
940 LINPUT "Input file name(1-6 char):",A$
960 Z$=TRIM$(A$)
980 ON ERROR GOTO 1060
1000 ENABLE KBD 0
1020 ASSIGN# 1 TO Z$
1040 GOTO 1540
1060 OFF ERROR @ CRT IS 1
1080 IF ERRN#67 THEN 760
1100 BEEP @ PRINT "*****ERROR - NO SUCH FILE"
```

```
1120 ENABLE KBD 1
1140 LINPUT "Put the program tape back and   hit 'END LINE'",A$
1160 GOTO 480
1180 !
1200 ! Project # 1,2.
1220 G1=1
1240 ENABLE KBD 0
1260 CLEAR
1280 ON ERROR GOTO 1460
1300 ASSIGN# 1 TO "FILE01"
1320 GOTO 1540
1340 G1=2
1360 ENABLE KBD 0
1380 CLEAR
1400 ON ERROR GOTO 1460
1420 ASSIGN# 1 TO "FILE02"
1440 GOTO 1540
1460 OFF ERROR @ BEEP
1480 IF G1=1 THEN PRINT "*****ERROR - ASSIGN 'FILE01'"
1500 IF G1=2 THEN PRINT "*****ERROR - ASSIGN 'FILE02'"
1520 GOTO 480
1540 OFF ERROR
1560 CLEAR @ CRT OFF
1580 READ# 1,1 ; P$,H$
1600 READ# 1,2 ; Y,D1,T1,D2,T2,G
1620 IF Y>0 THEN 1700
1640 BEEP @ PRINT "*****ERROR - NO DATA"
1660 ASSIGN# 1 TO *
1680 GOTO 480
1700 FOR I=1 TO Y
1720 READ# 1,I+2 ; T(I),D(I),F1()
1740 FOR J=1 TO 32
1760 F(I,J)=F1(J)
1780 NEXT J
1800 NEXT I
1820 ASSIGN# 1 TO *
1840 ENABLE KBD 1
1860 IF G1#0 THEN 2000
1880 !
1900 ! Put program tape back.
1920 CLEAR
1940 LINPUT "Put the program tape back and   hit 'END LINE'",A$
1960 !
1980 ! Get infor. for plot.
2000 CLEAR
2020 DISP USING 2040 ; G,P$
2040 IMAGE 5X,"*****PROJECT  ",D," *****",/,"Project ID:",/,K,2/
2060 DISP USING 2080 ; Y
2080 IMAGE "# of data points: ",DDD
2100 DISP USING 2120 ; HMS$(T1),MDY$(D1+MDY("12/31/1981"))
2120 IMAGE "Starting time and date:",/,K,10X,K
2140 DISP USING 2160 ; HMS$(T2),MDY$(D2+MDY("12/31/1981"))
```

```
2160 IMAGE "Ending time and date:",/,K,10X,K,2/
2180 IF C=0 THEN 2240
2200 LINPUT "Hit 'END LINE'",A$
2220 GOTO 3460
2240 ON ERROR GOTO 2000
2260 DISP "Enter time interval for plottingthe data:"
2280 LINPUT "mm/dd/yyyy hh:mm:ss    (start)",A$
2300 A$=TRIM$(A$)
2320 D3=MDY(A$[1,10])-MDY("12/31/1981")
2340 T3=HMS(TRIM$(A$[12]))
2360 LINPUT "mm/dd/yyyy hh:mm:ss    (end)",A$
2380 A$=TRIM$(A$)
2400 D4=MDY(A$[1,10])-MDY("12/31/1981")
2420 T4=HMS(TRIM$(A$[12]))
2440 OFF ERROR
2460 O=0
2480 IF D3>D2 THEN O=1
2500 IF D4<D1 THEN O=1
2520 IF D3>D4 THEN O=1
2540 IF D3#D4 THEN 2580
2560 IF T3=T4 THEN O=1
2580 IF O=0 THEN 2640
2600 BEEP @ PRINT "*****ERROR - WRONG INTERVAL"
2620 GOTO 2000
2640 CLEAR @ ON ERROR GOTO 2640
2660 DISP "YMIN,YMAX for plot";
2680 INPUT Y3,Y4
2700 LINPUT "Input LABEL for Y axis:",W$
2720 LINPUT "Set plotter and hit 'END LINE'",A$
2740 OFF ERROR
2760 !
2780 ! Draw axes and boundar.
2800 X1=D3+T3/86400-365
2820 X2=D4+T4/86400-365
2840 Y1=Y3 @ Y2=Y4
2860 PLOTTER IS 705
2880 LOCATE 14,100*RATIO-14,12,82
2900 SCALE X1,X2,Y1,Y2
2920 FXD 3,3 @ CSIZE 2,.5,0
2930 YAXIS X2
2940 LAXES
2960 MOVE X1,Y2 @ DRAW X2,Y2
2980 DRAW X2,Y1
3000 CSIZE 3,.5,0
3020 DEG @ LDIR 90
3040 SETGU @ MOVE 2,40
3060 LABEL W$
3080 LDIR 0
3100 MOVE 40,1
3120 LABEL "TIME   (DAYS SINCE 12/31/1982)"
3140 MOVE 14,95 @ CSIZE 4,.5,0
3160 LABEL USING 3180 ; P$
```

```
3180 IMAGE K
3200 MOVE 14,91 @ CSIZE 3,.5,0
3220 LABEL USING 3240 ; G+3
3240 IMAGE "INSTALLATION: ",D
3260 MOVE 14,89
3280 A$[1,8]=HMS$(T3) @ A$[9,13]="     " @ A$[14,23]=MDY$(D3+MDY("12/31
/1981"))
3300 LABEL USING 3320 ; A$[1,23]
3320 IMAGE "STARTING TIME AND DATE FOR PLOT:",5X,K
3340 MOVE 14,87
3360 A$[1,8]=HMS$(T4) @ A$[9,13]="     " @ A$[14,23]=MDY$(D4+MDY("12/31
/1981"))
3380 LABEL USING 3400 ; A$[1,23]
3400 IMAGE "ENDING TIME AND DATE FOR PLOT:",7X,K
3420 !
3440 ! Get infor. for curve.
3460 PEN 1
3480 I=0
3500 SETUU @ MOVE X1,Y1
3520 CLEAR
3540 LINPUT "Plot data(Y/N)?",A$
3560 IF A$="N" THEN 5000
3600 ON ERROR GOTO 3620
3620 LINPUT "Input channel # to be plotted:",A$
3640 J=VAL(A$)
3660 OFF ERROR
3680 IF J<32*(G-1)+1 THEN 3520
3700 IF J>32*G THEN 3520
3720 IF G=2 THEN J=J-32
3740 ON ERROR GOTO 3760
3760 LINPUT "Input # of points to bypass for identification:",A$
3780 N=VAL(A$)
3800 OFF ERROR
3820 IF N<0 THEN 3520
3840 IF N>=Y THEN 3980
3860 ON ERROR GOTO 3880
3880 LINPUT "Identification # (1-6)?",A$
3900 I=VAL(A$)
3920 IF I<1 THEN 3880
3940 IF I>6 THEN 3880
3960 OFF ERROR
3961 IF J<21 THEN 3980
3962 IF J>30 THEN 3980
3964 LINPUT "INPUT MODULUS; Ex10^6, E=1 FOR MICROSTRAIN OUTPUT",A$
3966 E=VAL(A$)
3980 LINPUT "Draw lines between points(Y/N)?",A$
4000 IF A$="Y" THEN 4100
4020 IF A$="N" THEN 4100
4040 GOTO 3980
4060 !
4080 ! Plot curve.
4100 M=0
```

```
4120 FOR L=1 TO Y-1
4140 X3=D(L)+T(L)/86400-365
4160 X4=D(L+1)+T(L+1)/86400-365
4180 IF X3<X1 THEN 4620
4200 IF X4>X2 THEN 4680
4201 Y3=F(L,J) @ Y4=F(L+1,J)
4202 IF J>30 THEN 4240
4203 IF J<21 THEN 4240
4205 Y3=Y3*E*1000000
4207 Y4=Y4*E*1000000
4220 GOTO 4240
4225 IF L-1<1 THEN 4232
4230 Y3=(F(L-1,J)+1*F(L,J)+F(L+1,J))*1000000/3
4231 GOTO 4234
4232 Y3=(F(L,J)*2+F(L+1,J))*1000000/3
4234 IF L+2>Y THEN 4238
4235 Y4=(F(L,J)+1*F(L+1,J)+F(L+2,J))*1000000/3
4236 GOTO 4240
4238 Y4=(F(L,J)+2*F(L+1,J))*1000000/3
4240 M=M+1
4260 IF M>N THEN M=0
4280 IF M#0 THEN 4580
4300 SETUU @ MOVE X3,Y3 @ SETGU
4320 ON I GOSUB 4340,4380,4420,4460,4500,4540
4330 GOTO 4580
4340 RPLOT .3,1,-1 @ RPLOT -.3,1,-1 @ RPLOT 0,0,2
4360 RETURN
4380 RPLOT .3,-1,-1 @ RPLOT -.3,-1,-1 @ RPLOT 0,0,2
4400 RETURN
4420 RPLOT -.5,.5,-1 @ RPLOT -.5,-.5,-1 @ RPLOT 0,0,2
4440 RETURN
4460 RPLOT .5,.5,-1 @ RPLOT .5,-.5,-1 @ RPLOT 0,0,2
4480 RETURN
4500 RPLOT -.5,0,1 @ RPLOT .5,0,-1 @ RPLOT 0,-.5,-2 @ RPLOT 0,.5,-1
4520 RETURN
4540 RPLOT -.5,-.5,1 @ RPLOT .5,.5,-1 @ RPLOT .5,-.5,-2 @ RPLOT -.5,.5
,-1
4560 RETURN
4580 SETUU @ IF A$#"Y" THEN 4620
4600 MOVE X3,Y3 @ DRAW X4,Y4
4620 NEXT L
4640 !
4660 ! Write legend.
4680 IF I=0 THEN 3500
4700 IF M1(I)=1 THEN 3480
4720 M1(I)=1
4740 K=8*(J-1)+1
4760 K1=K+7
4780 Q=.5
4800 IF I=1 THEN Q=0
4820 IF I=2 THEN Q=1
4840 SETGU @ MOVE 100*RATIO-26,80-I*2+Q
```

```
4860 ON I GOSUB 4340,4380,4420,4460,4500,4540
4880 MOVE 100*RATIO-24,80-I*2
4900 LABEL USING 4920 ; H$[K,K1]
4920 IMAGE "=",K
4940 GOTO 3480
4960 !
4980 ! Plot more?
5000 CLEAR
5020 PEN 0
5040 LINPUT "Continue this plot(Y/N)?",A$
5060 IF A$="N" THEN 5110
5080 IF A$#"Y" THEN 5040
5100 C=1 @ GOTO 480
5110 C=0
5111 FOR I=1 TO 6
5112 M1(I)=0
5113 NEXT I
5120 LINPUT "Plot more from this file(Y/N)?",A$
5140 IF A$="Y" THEN 2000
5160 IF A$="N" THEN 480
5180 GOTO 5120
5200 !
5220 ! Help.
5240 CLEAR @ ENABLE KBD 1
5260 DISP USING 5280
5280 IMAGE 9X,"*****HELP*****",2/
5300 DISP "The keys operate as follows:"
5320 DISP USING 5340
5340 IMAGE 1/,"K1= plot of 'PROJECT 1'",/,"K2= plot of 'PROJECT 2'"
5360 DISP USING 5380
5380 IMAGE "K3= plot of a 'STORED' file",/,"K4= 'HELP'",/,"K5= 'EXIT'
program 'PLOT'",6/
5400 LINPUT "Hit 'END LINE' to return",A$
5420 GOTO 480
5440 !
5460 ! Exit.
5480 CLEAR @ ENABLE KBD 33
5500 STOP
5520 END
```

MODULE 'MODATA'

```
1000 ! MODATA - VERSION 1   7/12/83
1010 CLEAR
1015 OPTION BASE 1
1020 INTEGER T1,F2,D7,P2,E,A7,M8,P0,09,F8,A,B,I,J,L,L0,L5,L8,Y9,R0,R1,R2,R3,K0,H

1030 DIM F$[18],O$[18],E$[100],G$[100],L$[96],N$[14],N0$[14],E0$[28],N9$[14],W0$
[22]
1040 DIM I$[4100],S$[100],K9$[480],K7$[96],09$[6],K$[100],B1$[100],P$[32],P1$[2]
,P9$[40],P7$[2]
1050 DIM R0$[32],R1$[96],R2$[2],L9$[2],C$[4],A$[18],Z$[1],Z1$[1],Z2$[1]
1055 DIM D4(32),F1(32)
1060 ON ERROR GOTO 1070 @ A=BTD("1") @ GOTO 1080
1070 DISP "No I/O rom" @ BEEP @ STOP
1080 S=0 @ ON ERROR GOTO 1100
1090 FOR I=3 TO 10 @ STATUS I,0 ; A@ IF A=2 THEN S=I @ I=10
1100 NEXT I
1110 IF S=0 THEN BEEP @ DISP "No modem" @ STOP
1120 T1=0 @ RESET S
1130 K9$="------------CONNECT-------------AUTO-ANS ANSWER   ORIG   STEST  "
1140 K9$=K9$&"terminal  DIAL   REDIAL  HANGUP ------------TERMINAL------------"
1150 K9$=K9$&" ESCAPE  BREAK   PRNT:OFF        direct transfer  frame  connect"
1160 K9$=K9$&"------------TRANSFER------------SHOWFILE  CAT                   "
1170 K9$=K9$&"terminal ENDXFER UPLOAD DOWNLOAD-------------FRAME--------------"
1180 K9$=K9$&" BITS:7    ODD     ERR:OFF        terminal ECHO:OFF FLIP:OFF CR    "
1190 K9$=K9$&"------------DIRECTORY------------ DELETE  RENAME   LIST    STATUS "
1200 K9$=K9$&"terminal  LOAD    ADD     STORE   "
1210 09$="OFFON " @ P9$="  NONE    ODD      EVEN  ALWAYS 1ALWAYS 0"
1220 Y9,09,F2,D7,X0,P2,E,C4,L,L0,L5,K0,F8,A7,M8,H=0 @ K$,P7$,P$,L$,E$,F$,O$=""
1230 P1=2 @ D8=4000 ! D8=LEN(I$)-100
1240 L9$=CHR$(13) @ E0$="Line >200 characters ignored" @ N9$="Name not found"
1250 W0$="Waiting for connection"
1260 ON ERROR GOTO 1270 @ LOADBIN "IPBIN"
1270 OFF ERROR @ CONTROL S,9 ; 225
1280 HALT S @ CONTROL S,5 ; 0@ CONTROL S,14 ; 19,17,0@ CONTROL S,11 ; 192
1285 STATUS S,9 ; A@ CONTROL S,9 ; BINIOR(A,128)
1290 IF M8=0 THEN K9=1 @ GOSUB 5300
1300 C$=CHR$(153)&CHR$(8)&CHR$(146)&CHR$(12)
1310 CONVERT KBD PAIRS ;  C$
1320 ON KBD GOSUB 1770
1330 IOBUFFER I$ @ IOBUFFER S$
1340 D=1
1350 SET TIMEOUT S;60000
1360 ON TIMEOUT S GOTO 1580
1370 ON ERROR GOSUB 1620
1380 IF M8 OR T1 THEN 1400
1390 DISP "Modcom ready for use" @ BEEP 20,30
1400 GOSUB 1720 @ ON INTR S GOTO 1590 @ M8=0 @ ENABLE INTR S;4 @ CONTROL S,5 ; 1
1
1410 TRANSFER S TO I$ INTR
1420 IF A7=0 THEN 1450 ELSE A7=0 @ A=(TIME+2) MOD 86400-2
1430 STATUS I$,1 ; B@ IF B#0 THEN 1450
```

```
1440 B=TIME @ IF B-2<A OR B-10000>A THEN 1430
1450 DISP "Connection established" @ K9=2 @ GOSUB 5300
1460 STATUS I$,0 ; A1,A2
1470 GOSUB 5980 @ IF M8 THEN 1590
1480 IF A2<A1 OR A2<D THEN 1460
1490 ON ERROR GOSUB 1620
1500 B1$=I$[D,A2] @ CDISP B1$ @ P7$=B1$[MAX(1,LEN(B1$)-1),LEN(B1$)] @ D=A2+1
1510 IF NOT P2 THEN 1540
1520 ON ERROR GOSUB 1700
1530 IF P1>2 THEN OUTPUT P1 USING "#,K" ; B1$ ELSE CPRINT B1$
1540 ON ERROR GOSUB 1620
1550 STATUS I$,1 ; A2
1560 IF A2<D8 THEN 1460
1570 IF D7 THEN GOSUB 3800 @ GOTO 1460 ELSE GOSUB 1750 @ GOTO 1460
1580 DISP "Time out" @ OFF TIMEOUT S @ OFF KBD @ OFF EOT S @ RESET S @ GOTO 1130

1590 HALT S @ OFF KBD @ T1=1 @ L=0 @ IF D7 THEN GOSUB 3120
1600 CONTROL S,1 ; 0@ IF M8 THEN 1280 ELSE CONTROL S,5 ; 0
1610 DISP "Modem carrier lost" @ BEEP 300,200 @ GOSUB 5860 @ GOTO 1280
1620 I$="" @ GOSUB 3950
1630 IF ERRN>=80 AND ERRN<=92 THEN DISP "Illegal entry" @ GOTO 1690
1640 IF ERRN>=60 AND ERRN<80 THEN DISP "File problem" @ GOTO 1690
1650 IF ERRN>=30 AND ERRN<60 THEN DISP "Program input buffer overflow" @ GOTO 16
90
1660 IF ERRN<30 THEN DISP "System error";ERRN;"at line";ERRL @ GOTO 1690
1670 IF ERRN>=123 AND ERRN<=131 THEN DISP "Mass storage/printer error";ERRN;"at"
;ERRL @ GOTO 1690
1680 DISP "Input buffer over-run ! Data lost!" @ OFF EOT S @ GOSUB 3988
1690 BEEP @ ON ERROR GOSUB 1620 @ RETURN
1700 DISP "Bad select code" @ DISP "Printer off" @ P2=NOT P2 @ K9$[151,153]="OFF
"
1710 IF K9=2 THEN GOSUB 5300 @ GOTO 1690 ELSE GOTO 1690
1720 STATUS S,3 ; A@ IF A MOD 2=0 THEN 1720
1730 OFF TIMER# 3 @ STATUS S,9 ; A@ CONTROL S,9 ; A+4
1740 C4=1 @ S$="" @ IF LEN(L$)=0 OR L5=0 THEN RETURN ELSE L=1 @ L0=1 @ L5=0 @ RE
TURN
1750 STATUS I$,1 ; A2@ I$=I$[D,A2]
1760 D=1 @ GOTO 3960
1770 K$=KBD$
1780 J0=FIND(K$) @ IF J0=0 THEN K=LEN(K$) @ GOTO 2480
1790 N=NUM(K$[J0,J0])
1800 IF N=154 THEN K$=K$[1,J0-1]&L9$&K$[J0+1] @ GOTO 1780
1810 IF N<=135 THEN ON K9 GOTO 5330,1970,2070,2140,2330
1820 IF N=142 THEN K$[J0,J0]=CHR$(19) @ GOTO 1780 ! PAUSE (XOFF)
1830 IF N=143 THEN K$[J0,J0]=CHR$(17) @ GOTO 1780 ! CONT (XON)
1840 IF N=158 THEN CLINE CLPOS+1 @ CCURSOR CCPOS+32 @ GOTO 2460 ! Roll Up
1850 IF N=159 THEN CLINE CLPOS-1 @ CCURSOR CCPOS-32 @ GOTO 2460 ! Roll Down
1860 IF N=150 THEN GOSUB 5300 @ GOTO 2470
1870 IF N=138 THEN PRINT @ GOTO 2470 ! Paper Advance
1880 IF N=139 THEN BEEP @ RESET S @ OFF KBD @ CLEAR @ STOP ! Reset
1890 IF N=165 THEN CCURSOR CLPOS*32 @ GOTO 2470 ! Home
1900 IF N=161 THEN CCURSOR CCPOS-32 @ GOTO 2440 ! Up arrow
```

```
1910 IF N=162 THEN CCURSOR CCPOS+32 @ GOTO 2440 ! Down arrow
1920 IF N=156 THEN CCURSOR CCPOS-1 @ GOTO 2440 ! Left arrow
1930 IF N=157 THEN CCURSOR CCPOS+1 @ GOTO 2440 ! Right arrow
1940 IF N=173 THEN OFF TIMER# 3 @ L=0 @ L0=0 @ GOTO 2440 ! SCRATCH
1950 IF N=141 THEN L0=1 @ GOTO 2440 ! RUN
1960 GOTO 2410
1970 ! soft keys p2
1980 IF N=133 THEN REQUEST S;1 @ GOTO 2470 ! Send BREAK
1990 IF N=129 THEN K9=3 @ GOSUB 5300 @ GOTO 2470 ! TRANSFER
2000 IF N=131 THEN K9=1 @ GOSUB 5300 @ GOTO 2470 ! CONNECT
2010 IF N=128 THEN K9=5 @ GOSUB 5300 @ GOTO 2470 ! DIRECT
2020 IF N=132 THEN K$[J0,J0]=CHR$(27) @ GOTO 1780 ! <escape>
2030 IF N=130 THEN K9=4 @ GOSUB 5300 @ GOTO 2470 ! FRAME
2040 IF N=134 THEN GOSUB 4230 @ GOTO 2470 ! PRINTER
2050 GOTO 2410
2060 ! soft keys p3
2070 IF N=128 THEN K9=2 @ GOSUB 5300 @ GOTO 2470 ! terminal
2080 IF N=133 THEN GOSUB 2650 @ GOTO 2470 ! CAT
2090 IF N=129 THEN GOSUB 3120 @ GOTO 2470 ! End transfer
2100 IF N=130 THEN 3190 ! Upload
2110 IF N=131 THEN GOSUB 3750 @ GOTO 2470 ! Download
2120 IF N=132 THEN 4050 ! showfile
2130 GOTO 2410
2140 ! softkeys p4
2150 IF N=128 THEN K9=2 @ GOSUB 5300 @ GOTO 2470 ! terminal
2160 IF N=129 THEN E=NOT E @ K9$[367,369]=O9$[E*3+1,E*3+3] @ GOSUB 5300 @ GOTO 2
310 ! ECHO
2170 IF N=130 THEN F8=NOT F8 @ K9$[376,378]=O9$[F8*3+1,F8*3+3] @ GOSUB 5300 @ GO
TO 2310 ! FLIP
2180 IF N#131 THEN 2220
2190 IF LEN(L9$)=1 THEN L9$=L9$&CHR$(10) @ K9$[382,384]="/LF" @ GOTO 2210
2200 L9$=L9$[1,1] @ K9$[382,384]="   "
2210 GOSUB 5300 @ GOTO 2310 ! CR - CR/LF
2220 IF N#132 THEN 2240 ELSE HALT S @ STATUS S,4 ; A@ A=BINEOR(A,1) @ CONTROL S,
4 ; A
2230 K9$[327,327]=VAL$(A MOD 4+5) @ GOSUB 5300 @ GOTO 2290 ! BITS
2240 IF N#133 THEN 2270 ELSE HALT S @ STATUS S,4 ; A@ B=A\8 MOD 8 @ B=BINIOR(B+1
,1) MOD 9
2250 A=BINIOR(BINAND(A,199),B*8) @ CONTROL S,4 ; A@ B=(B+1)\2
2260 K9$[330,337]=P9$[B*8+1,B*8+8] @ GOSUB 5300 @ GOTO 2290 ! PARITY
2270 IF N#134 THEN 2410 ELSE HALT S @ STATUS S,9 ; A@ B=(BINAND(A,8)+4) MOD 11
2280 CONTROL S,9 ; BINEOR(A,8)@ K9$[343,345]=O9$[B,B+2] @ GOSUB 5300 ! ERRCHK
2290 IF C4=0 THEN 2470
2300 HALT S @ TRANSFER S TO I$ INTR @ STATUS S$,0 ; A,B@ IF A<=B THEN TRANSFER S
$ TO S INTR
2310 IF K9=4 THEN O$="" @ GOTO 2470 ELSE 2470
2320 ! softkeys p5
2330 IF N=128 THEN K9=2 @ GOSUB 5300 @ GOTO 2470 ! TERMINAL
2340 IF N=129 THEN GOSUB 4450 @ GOTO 2430 ! LOAD
2350 IF N=131 THEN GOSUB 5120 @ GOTO 2430 ! REPLACE
2360 IF N=134 THEN GOSUB 5290 @ GOSUB 5170 @ GOTO 2430 ! LIST
2370 IF N=135 THEN GOSUB 2710 @ GOTO 2470 ! STATUS
```

```
2380 IF N=132 THEN GOSUB 5250 @ GOTO 2430
2390 IF N=133 THEN GOSUB 5290 @ GOSUB 4510 @ GOTO 2430 ! RENAME
2400 IF N=130 THEN GOSUB 5290 @ GOSUB 4590 @ GOTO 2430 ! ADD
2410 BEEP 125,100 @ BEEP 60,150 @ GOTO 2470
2420 BEEP 150,75 @ GOSUB 5290 @ GOTO 2470
2430 IF O9=0 THEN 2290 ELSE ASSIGN# 2 TO * @ O9=0 @ GOTO 2290
2440 IF (CCPOS\32+1) MOD 64=CLPOS THEN CCURSOR CCPOS+512
2450 IF CCPOS\32=(CLPOS+16) MOD 64 THEN CCURSOR CCPOS-512
2460 K7$="" @ CDISP K7$
2470 ON ERROR GOTO 2480 @ K$=K$[1,JO-1]&K$[JO+1] @ GOTO 1780
2480 GOSUB 2490 @ IF K$#"" THEN 1780 ELSE ON KBD GOSUB 1770 @ RETURN
2490 ON ERROR GOSUB 1620 @ IF F8=0 THEN 2530
2500 FOR X=1 TO K @ A=NUM(K$[X,X]) @ IF (A>64 AND A<91 OR A>96 AND A<123)=0 THEN
 2520
2510 K$[X,X]=CHR$(BINEOR(A,32))
2520 NEXT X
2530 IF NOT K THEN 2640
2540 OUTPUT S$ USING "#,K" ; K$[1,K]
2550 RESUME S
2560 STATUS S$,3 ; A3@ STATUS S$,1 ; A1
2570 IF NOT A3 AND A1 THEN TRANSFER S$ TO S INTR
2580 IF NOT E THEN 2640
2590 K7$=K$[1,K] @ CDISP K7$
2600 ON ERROR GOSUB 1700
2610 IF P2#0 AND P1>2 THEN OUTPUT P1 USING "#,K" ; K$
2620 IF P2#0 AND P1=2 THEN CPRINT K$
2630 ON ERROR GOSUB 1620
2640 K$=K$[K+1,LEN(K$)] @ RETURN
2650 ON ERROR GOTO 3055 @ CLEAR @ CAT @ GOSUB 5300 @ RETURN
2710 PRINT "Modcom status:" @ PRINT @ PRINT "Name............"&O$;@ IF O$="" THE
N PRINT "NONE" ELSE PRINT
2720 PRINT "Phone number...."&P$;@ IF P$="" THEN PRINT "NONE" ELSE PRINT
2730 PRINT "Logon sequence.."&L$;@ IF L$="" THEN PRINT "NONE" ELSE PRINT
2740 PRINT "Echo............";@ IF E THEN PRINT "ON" ELSE PRINT "OFF"
2750 PRINT "Flip............"&K9$[376,378] @ PRINT "Data bits......."&K9$[327,32
7]
2760 J=330 @ IF K9$[J,J]=" " THEN J=332
2770 PRINT "Parity.........."&K9$[J,337] @ PRINT "Error check....."&K9$[343,345]

2780 PRINT "EOL.............";
2790 FOR I=1 TO LEN(L9$)
2800 IF L9$[I,I]=CHR$(10) THEN PRINT "/LF";@ GOTO 2830
2810 IF L9$[I,I]=CHR$(13) THEN PRINT "CR";@ GOTO 2830
2820 PRINT NUM(L9$[I,I]);
2830 NEXT I @ PRINT @ PRINT @ PRINT @ PRINT @ PRINT @ RETURN
2840 GOSUB 3040 @ ON ERROR GOTO 3010 @ DISP "File name";@ INPUT F$
2850 IF F$="" THEN 3030
2860 ON ERROR GOTO 2990
2870 ASSIGN# 1 TO F$ @ DISP "File open" @ XO=1 @ I$="" @ GOSUB 3950 @ RETURN
2880 GOSUB 3040 @ ON ERROR GOTO 3020
2890 D7=0 @ DISP "Create a file :Y/N";@ INPUT B1$
2900 IF B1$="" THEN BEEP 150,75 @ GOTO 3030
```

```
2910 B1$=UPC$(B1$) @ F7=B1$[1,1]#"Y" @ IF B1$[1,1]#"N" AND F7 THEN BEEP @ GOTO 2
880
2920 IF F7 THEN 2840
2930 ON ERROR GOTO 3050
2940 DISP "File size in 256 byte records";@ INPUT B1$@ IF B1$="" THEN 3030
2950 I1=VAL(B1$) @ ON ERROR GOTO 2970
2960 CREATE F$,I1 @ DISP "File created" @ GOTO 2860
2970 GOSUB 3060 @ DISP "File not created"
2980 BEEP @ I$="" @ GOSUB 3950 @ GOTO 2930
2990 GOSUB 3060 @ BEEP @ WAIT 1200
3000 IF D4 THEN 2890 ELSE 2840
3010 F$=""
3020 DISP "Bad input" @ BEEP @ WAIT 1200 @ GOTO 2840
3030 F$="" @ BEEP 150,75
3040 GOSUB 5290 @ I$="" @ GOSUB 3950 @ RETURN
3050 DISP "Bad input" @ BEEP @ GOTO 2930
3055 GOSUB 5300 @ BEEP
3060 IF ERRN=63 THEN DISP "Duplicate file name" @ RETURN
3070 IF ERRN=61 OR ERRN=65 OR ERRN=128 OR ERRN=124 THEN DISP "Storage medium ful
l" @ RETURN
3080 IF ERRN=62 THEN DISP "Cartridge out" @ RETURN
3090 IF ERRN=130 THEN DISP "Disc out" @ RETURN
3095 IF ERRN=89 THEN DISP "Bad size" @ RETURN
3100 DISP "Bad file name" @ RETURN
3110 ! End transfer
3120 IF NOT D7 THEN DISP "No transfer!" @ BEEP @ GOSUB 5300 @ RETURN
3130 IF F2 THEN 3680
3140 GOSUB 3890 @ ON ERROR GOTO 3690
3150 D7=0 @ PRINT# 1 ; "" @ GOTO 3680
3160 GOSUB 5300
3170 ON KBD GOSUB 1770 @ RETURN
3180 ! Upload
3190 IF C4=0 THEN GOSUB 3720 @ GOTO 3160
3200 IF D7 THEN GOSUB 3730 @ GOTO 3160 ELSE D4=0 @ GOSUB 2840
3210 IF F$="" THEN D7=0 @ GOTO 3170
3220 IF D7 OR NOT X0 THEN DISP "No transfer - File not opened" @ BEEP @ GOTO 316
0
3230 D7=1 @ B1$="" @ F2=1 @ F3=0 ! file xfer on
3240 ON ERROR GOTO 3300
3250 GOSUB 5290
3260 DISP "Does your host prompt:Y/N";
3270 INPUT A$@ A$=UPC$(A$) @ P9=0
3280 IF A$="" THEN D7=0 @ F2=0 @ BEEP 150,75 @ CLEAR @ GOTO 3160
3290 IF A$[1,1]="Y" OR A$[1,1]="N" THEN 3310
3300 BEEP @ GOTO 3250
3310 IF A$[1,1]="Y" THEN P9=1
3320 IF P9=1 THEN 3360 ELSE Y9=0
3325 ON ERROR GOTO 3660
3330 DISP "Enter delay (millisecs) between sending lines" @ INPUT A$
3340 IF A$="" THEN 3280
3350 Y9=VAL(A$) @ GOTO 3410
3360 WAIT 500
```

```
3370 ! SAVE PROMPT CHARACTER
3380 X=LEN(P7$) @ IF X=0 THEN DISP "No prompt char" @ GOTO 3260 ELSE Z$=P7$[X,X]

3390 X=X-1 @ IF X=0 THEN Z1$="" ELSE Z1$=P7$[X,X]
3400 ! get a line
3410 DISP "Upload in progress" @ ON KBD GOSUB 1770
3420 IF NOT D7 THEN 3680
3425 GOTO 7000
3430 OFF ERROR @ GOTO 3490
3435 CDISP I$ @ I$="" @ GOSUB 3950
3440 IF F3=0 THEN 3490
3450 F3=0 @ RETURN
3490 OUTPUT S$ USING "#,K" ; B1$
3500 OUTPUT S$ USING "#,K" ; E$
3510 IF E THEN CDISP B1$ @ CDISP E$
3520 TRANSFER S$ TO S INTR
3530 STATUS S$,3 ; A3@ IF A3 THEN 3530
3540 IF F9#1 THEN WAIT Y9
3550 RETURN
3560 X=LEN(I$) @ IF X=0 THEN 3560 ELSE Z2$=I$[X,X]
3570 IF Z2$=Z$ THEN 3590
3580 IF Z2$<>Z1$ THEN 3560
3590 WAIT 500 @ GOTO 3430
3600 IF ERRN=56 THEN DISP E0$ @ BEEP @ GOTO 3490
3610 F2=0 @ IF ERRN=71 OR 72 THEN 3680
3620 D7=0
3630 IF ERRN=66 THEN DISP "File not opened ! ";
3640 DISP "No file transfer" @ BEEP @ X0=0 @ RETURN
3660 DISP "Bad input...Please reenter" @ BEEP @ WAIT 1200
3670 GOSUB 5290 @ GOTO 3325
3680 D7=0 @ X0=0 @ ASSIGN# 1 TO * @ DISP "File transfer complete" @ GOSUB 5300
3690 IF NOT F2 THEN 3710
3700 ON ERROR GOTO 3710 @ F2=0 @ F3=1 @ BEEP 150,75 @ RETURN
3710 BEEP 150,75 @ GOSUB 5300 @ RETURN
3720 GOSUB 5290 @ BEEP @ DISP "No connection!" @ RETURN
3730 DISP "Transfer already in progress" @ BEEP @ RETURN
3740 ! Download
3750 IF C4=0 THEN 3720
3800 ON ERROR GOSUB 3980
3835 WAIT 30 @ HALT S @ STATUS S,9 ; A@ CONTROL S,9 ; BINIOR(A,1)
3839 TRANSFER S TO I$ INTR
3840 STATUS S$,3 ; A3@ IF A3 THEN 3840
3850 OUTPUT S$ USING "#,K" ; CHR$(17)
3860 RESUME S
3870 TRANSFER S$ TO S INTR
3880 RETURN
3890 ON ERROR GOSUB 3980
3900 STATUS I$,0 ; A1,A2@ P=POS(I$[A1,A2],CHR$(10)) @ IF P=0 THEN 3930
3910 ENTER I$ ; B1$@ IF B1$="" THEN B1$=" "
3920 PRINT# 1 ; B1$ @ GOTO 3900
3930 IF D<=A2+1 THEN B1$=I$[D,A2] @ CDISP B1$
3940 I$=I$[A1,A2]
```

```
3950 D=LEN(I$)+1
3960 CONTROL I$,0 ; 1@ A1=1 @ A2=0
3970 RETURN
3980 DISP @ IF ERRN=72 THEN DISP "File too small!" @ I$="" @ GOSUB 3950 @ F2=0 @
 GOTO 3680
3983 IF ERRN=114 THEN DISP "Input buffer overflow, data lost" ELSE DISP "Error";
ERRN;"on line";ERRL
3985 GOSUB 3120 @ ON ERROR GOSUB 3980 @ I$="" @ GOSUB 3950
3988 STATUS S,9 ; A@ CONTROL S,9 ; A+4@ RETURN
3990 ! EOT
4000 STATUS I$,2 ; I
4010 IF I#0 THEN ON EOT S GOSUB 4000 @ RETURN
4020 STATUS S,11 ; U0@ G$=I$ @ I$="" @ CONTROL I$,0 ; 1,0
4030 IF U0=1 OR U0=65 THEN RETURN ELSE E$=G$ @ RETURN
4040 ! show file
4050 IF D7 THEN GOSUB 3730 @ GOTO 3160 ELSE D7,F2=1
4060 I$="" @ GOSUB 3950 @ D4=0 @ GOSUB 2840 @ IF F$="" THEN D7=0 @ GOTO 3160 ELS
E ON KBD GOSUB 1770
4070 ON ERROR GOTO 4140
4080 IF NOT D7 THEN 4210
4090 READ# 1 ; B1$
4100 IF B1$="" THEN 4210
4110 DISP B1$ @ IF NOT P2 THEN 4080 ELSE ON ERROR GOSUB 1700
4120 IF P1>2 THEN OUTPUT P1 ;B1$ ELSE PRINT B1$
4130 ON ERROR GOTO 4140 @ GOTO 4080
4140 IF ERRN=71 OR ERRN=72 THEN 4210
4150 IF ERRN=66 THEN DISP "File not opened" @ GOTO 4210
4160 IF ERRN=64 THEN DISP "Empty file" @ GOTO 4210
4170 IF ERRN=62 OR ERRN=130 THEN GOSUB 3080 @ GOTO 4210
4180 IF ERRN=56 THEN DISP E0$ @ BEEP @ GOTO 4090
4190 IF ERRN>=60 AND ERRN<80 THEN DISP "File in error" @ GOTO 4210
4200 IF ERRN>=123 AND ERRN<=131 THEN DISP "File/printer"
4210 BEEP 150,75 @ B1$,F$="" @ DISP "*** End of file ***" @ F2,D7=0 @ RETURN
4220 ! printer
4230 P2=NOT P2 @ K9$[151,153]=O9$[P2*3+1,P2*3+3] @ IF NOT P2 THEN GOSUB 5300 @ G
OTO 4360
4240 GOSUB 5290 @ DISP "Old select code is ";P1 @ DISP "New select code";
4250 ON ERROR GOTO 4370 @ INPUT A$@ IF A$="" THEN GOSUB 5290 @ GOTO 4360
4260 I1=VAL(A$) @ IF I1<1 THEN 4340
4270 IF I1<99 THEN B=2 @ A=I1 ELSE B=1 @ A=I1\100
4280 IF A>10 THEN 4340
4290 IF (A=1 OR A=2) AND B=2 THEN 4350
4300 IF A<3 THEN 4340
4310 IF RIO(A,0)=255 THEN 4340
4320 IF A=S THEN 4340
4330 GOTO 4350
4340 BEEP @ DISP "Invalid select code" @ WAIT 1200 @ GOTO 4240
4350 GOSUB 5290 @ P1=I1
4360 RETURN
4370 DISP "Bad input" @ BEEP @ WAIT 1200 @ GOTO 4240
4380 ON ERROR GOTO 4410 @ R=1 @ ASSIGN# 2 TO "PHONE#" @ O9=1
4390 READ# 2,R ; N$@ IF N$=N0$ THEN A=1 @ RETURN
```

```
4400 IF N$#CHR$(255) THEN R=R+1 @ GOTO 4390 ELSE A=0 @ RETURN
4410 IF ERRN=72 OR ERRN=71 THEN A=0 @ RETURN
4420 IF ERRN=67 THEN DISP "Phone directory not found" @ A=0 @ RETURN
4430 A=0 @ GOTO 3055
4440 READ# 2 ; P$,L$,P0,B,L9$,F8,E@ GOSUB 5200 @ O$=N$ @ DISP "Loaded" @ RETURN
4450 ! LOAD
4460 GOSUB 5290 @ ON ERROR GOTO 4460 @ DISP "Name";@ INPUT N0$@ IF N0$="" THEN 4
990
4470 GOSUB 4380 @ IF A=0 THEN DISP N9$ @ WAIT 1200 @ GOTO 4460
4480 GOSUB 4440 @ L=0 @ RETURN
4490 ! RENAME
4500 WAIT 1200 @ GOSUB 5290
4510 ON ERROR GOTO 4500 @ DISP "Name to change";@ INPUT N0$@ IF N0$="" THEN 4990

4520 GOSUB 4380 @ IF A=0 THEN DISP N9$ @ GOTO 4500 ELSE B=R
4530 ON ERROR GOTO 4530 @ DISP "New name";@ INPUT N0$@ IF N0$="" THEN 4990
4540 GOSUB 4380 @ IF A#0 THEN DISP "Name already exists" @ GOTO 4500
4550 READ# 2,B ; R0$,R0$,R1$,R0,R1,R2$,R2,R3@ PRINT# 2,B ; N0$,R0$,R1$,R0,R1,R2$
,R2,R3
4560 DISP "Name changed" @ RETURN
4570 ! ADD
4580 WAIT 1200 @ GOSUB 5290
4590 ON ERROR GOTO 4590 @ DISP "New name";
4600 INPUT N0$@ IF N0$="" THEN 4990
4610 ON ERROR GOTO 5000 @ N$=N0$ @ ASSIGN# 2 TO "PHONE#" @ A,R=0 @ O9=1
4620 R=R+1 @ READ# 2,R ; B1$@ IF B1$=N$ THEN DISP "Name already exists" @ GOTO 4
580
4630 IF B1$[1,1]=CHR$(128) AND A=0 THEN A=R @ GOTO 4620
4640 IF B1$[1,1]#CHR$(255) THEN 4620
4650 IF A=0 THEN A=R
4660 R=A
4670 CLEAR
4680 DISP "Current phone:" @ IF P$="" THEN DISP "NONE" ELSE DISP P$
4690 ON ERROR GOTO 4915 @ DISP "New phone";@ INPUT R0$@ IF R0$="" THEN R0$=P$
4700 DISP "Current logon:" @ IF L$="" THEN DISP "NONE" ELSE DISP L$
4710 DISP "New logon (N for none)";@ LINPUT R1$@ IF R1$="" THEN R1$=L$
4720 IF R1$="N" OR R1$="n" THEN R1$=""
4730 HALT S @ STATUS S,4 ; P0@ STATUS S,9 ; B
4740 K7$="OFF@ON@" @ DISP "Current echo:";@ IF E THEN DISP "ON" ELSE DISP "OFF"
4750 DISP "New echo";@ R0=FNI(E) @ IF R0=-1 THEN 4750
4760 DISP "Current flip:";@ IF F8 THEN DISP "ON" ELSE DISP "OFF"
4770 DISP "New flip";@ R1=FNI(F8) @ IF R1=-1 THEN 4770
4780 K7$="7@8@" @ A=BINAND(P0,3)-2 @ DISP "Current data bits:";VAL$(A+7)
4790 DISP "New bits";@ R2=FNI(A) @ IF R2=-1 THEN 4790 ELSE P0=BINAND(P0,252)+R2+
2
4800 K7$="NONE@ODD@EVEN@ALWAYS 1@ALWAYS 0@" @ DISP "Current parity:";@ A=P0\8 MO
D 8
4810 A=(A+1)\2 @ DISP K9$[332-A\3*2,337]
4820 DISP "New parity";@ R2=FNI(A) @ IF R2=-1 THEN 4820
4830 A=R2*2-1 @ IF A=-1 THEN A=0
4840 P0=BINAND(P0,199)+A*8
4850 DISP "Current error check:";@ B=BINAND(B,8)\8 @ IF B=1 THEN DISP "ON" ELSE
```

```
DISP "OFF"
4860 K7$="OFF@ON@"
4870 DISP "New error check";@ A=FNI(B) @ IF A=-1 THEN 4870 ELSE B=A
4880 DISP "Current EOL:";@ IF LEN(L9$)=2 THEN DISP "CR/LF" @ R3=1 ELSE DISP "CR"
 @ R3=0
4890 K7$="CR@CR/LF@" @ DISP "New EOL";@ A=FNI(R3) @ IF A=-1 THEN 4890
4900 IF A=1 THEN R2$=CHR$(13)&CHR$(10) ELSE R2$=CHR$(13)
4910 PRINT# 2,R ; N$,R0$,R1$,P0,B*8,R2$,R1,R0 @ DISP "Saved" @ GOSUB 5300 @ RETU
RN
4915 BEEP @ DISP "Phone number too long" @ WAIT 1200 @ GOTO 4690
4920 DEF FNI(X)
4930 B1$="" @ INPUT B1$@ B1$=UPC$(B1$) @ IF B1$="" THEN FNI=X @ GOTO 4980
4940 I1=1 @ I,I0=0
4950 IF I0=LEN(K7$) THEN FNI=-1 @ GOTO 4980
4960 I0=POS(K7$[I1,LEN(K7$)],"@")+I1-1 @ IF I0=0 THEN FNI=-1 @ GOTO 4980
4970 IF K7$[I1,I0-1]=B1$ THEN FNI=I @ GOTO 4980 ELSE I=I+1 @ I1=I0+1 @ GOTO 4950

4980 FN END
4990 BEEP 150,75 @ GOSUB 5290 @ RETURN
5000 IF ERRN#67 AND ERRN#72 AND ERRN#71 THEN GOTO 3055
5005 IF ERRN=71 THEN PRINT# 2,R ; CHR$(255) @ R=R-1 @ GOTO 4620
5010 IF ERRN=72 THEN 5050 ELSE DISP "Phone directory doesn't exist."
5020 DISP "Do you want to create it(Y/N)";@ INPUT B1$@ IF B1$="" THEN BEEP 150,7
5 @ GOTO 5290
5025 B=POS("YN",UPC$(B1$[1,1])) @ IF B=0 THEN 5010
5030 IF B=2 THEN 5290 ELSE CREATE "PHONE#",20,164 @ ASSIGN# 2 TO "PHONE#"
5040 FOR B=1 TO 20 @ PRINT# 2,B ; CHR$(255) @ NEXT B @ A,R=0 @ O9=1 @ GOTO 4620
5050 IF A#0 THEN 4660 ELSE DISP "Directory is full, expanding" @ DISP "Please wa
it"
5060 A=R+19 @ CREATE CHR$(254),A,164 @ ASSIGN# 3 TO CHR$(254) @ RENAME "PHONE#"
TO CHR$(255)
5070 FOR B=1 TO R-1 @ READ# 2,B ; N0$,B1$,K7$,P0,I,P1$,J,O9
5080 PRINT# 3,B ; N0$,B1$,K7$,P0,I,P1$,J,O9 @ NEXT B @ FOR B=R TO A @ PRINT# 3,B
 ; CHR$(255)
5090 NEXT B @ ASSIGN# 2 TO * @ ASSIGN# 3 TO * @ PURGE CHR$(255)
5100 RENAME CHR$(254) TO "PHONE#" @ ASSIGN# 2 TO "PHONE#" @ O9=1 @ A=R @ GOTO 46
70
5110 ! STORE
5120 GOSUB 5290 @ ON ERROR GOTO 5130
5130 DISP "Name to store";@ INPUT N0$@ IF N0$="" THEN 4990
5140 GOSUB 4380 @ IF A=0 THEN DISP N9$ @ WAIT 1200 @ GOTO 5120
5150 GOTO 4670
5160 ! LIST
5170 ON ERROR GOTO 5193 @ ASSIGN# 2 TO "PHONE#" @ R=0 @ O9=1 @ DISP "Names:"
5180 R=R+1 @ READ# 2,R ; B1$@ IF B1$=CHR$(255) THEN 5197
5190 IF B1$=CHR$(128) THEN 5180 ELSE DISP B1$ @ GOTO 5180
5193 IF ERRN#72 AND ERRN#71 THEN 3055
5197 DISP "End of directory" @ RETURN
5200 HALT S @ CONTROL S,4 ; BINAND(P0,63)@ K9$[327,327]=VAL$(P0 MOD 4+5)
5210 A=(P0\8 MOD 8+1)\2 @ K9$[330,337]=P9$[A*8+1,A*8+8] @ STATUS S,9 ; A@ A=BINA
ND(A,247)
5220 CONTROL S,9 ; BINIOR(A,B)@ B=B\8 @ K9$[343,345]=O9$[B*3+1,B*3+3]
```

```
5230 IF LEN(L9$)=2 THEN K9$[382,384]="/LF" ELSE K9$[382,384]="   "
5240 K9$[376,378]=O9$[F8*3+1,F8*3+3] @ K9$[367,369]=O9$[E*3+1,E*3+3] @ RETURN
5250 ! DELETE
5260 ON ERROR GOTO 5260 @ GOSUB 5290 @ DISP "Name to delete";@ INPUT NO$@ IF NO$
="" THEN 4990
5270 GOSUB 4380 @ IF A=0 THEN DISP N9$ @ WAIT 1200 @ GOTO 5260
5280 PRINT# 2,R ; CHR$(128) @ DISP "Deleted" @ RETURN
5290 CLEAR
5300 J=CCPOS\32-CLPOS @ IF J<0 THEN J=J+64
5310 B=CCPOS @ J=J-12 @ IF J>0 THEN CLINE CLPOS+J
5320 CCURSOR CLPOS*32+32*13 @ K7$=K9$[K9*96-95,K9*96] @ CWRITE K7$ @ CCURSOR B @
 RETURN
5330 ON N-127 GOTO 5880,5340,5950,5810,5560,5670,5540,5720
5340 ! DIAL
5350 GOSUB 5960 @ IF H THEN 2470
5360 OFF TIMER# 3 @ S$=P$ @ GOSUB 5930 @ IF P$="" THEN P$=S$ @ GOTO 2420 ELSE P1
$=P$[1,1]
5370 IF P1$="@" OR P1$>="0" AND P1$<="9" THEN L5=0 @ L$,O$="" @ GOTO 5400
5380 NO$=P$[1,MIN(14,LEN(P$))] @ GOSUB 4380 @ IF A#1 THEN DISP N9$ @ P$=S$ @ WAI
T 1200 @ GOTO 5350
5390 GOSUB 4440 @ L5=1
5400 IF P$#"" THEN GOSUB 5410 @ GOTO 5530 ELSE DISP "No phone number" @ BEEP @ W
AIT 1200 @ GOSUB 5290 @ GOTO 2470
5410 B1$="Dialing " @ CWRITE B1$ @ K0=0 @ K$="!" @ ON KBD GOSUB 5500
5420 CONTROL S,2 ; 6@ WAIT 2000 @ H=1
5430 FOR I=1 TO LEN(P$) @ IF K0 THEN RETURN
5440 P1$=P$[I,I]
5450 IF P1$="@" THEN CWRITE P1$ @ WAIT 2000 @ GOTO 5490
5460 K=NUM(P1$)-48 @ IF K<0 OR K>9 THEN CWRITE P1$ @ GOTO 5490
5470 CWRITE P1$ @ IF K=0 THEN K=10
5480 FOR J=1 TO K @ ASSERT S;5 @ ON TIMER# 3,51 GOTO 5482
5481 GOTO 5481
5482 ASSERT S;6 @ ON TIMER# 3,25 GOTO 5484
5483 GOTO 5483
5484 OFF TIMER# 3 @ NEXT J @ WAIT 700
5490 NEXT I @ DISP @ ON KBD GOSUB 5520 @ RETURN
5500 K$=KBD$ @ IF K$=CHR$(131) OR K$=CHR$(139) THEN K0=1 @ K$=K$&K$ @ RETURN
5510 ON KBD GOSUB 5500 @ RETURN
5520 K$=K$&KBD$ @ RETURN
5530 IF K0 THEN 2470 ELSE 5550
5540 GOSUB 5960 @ IF H THEN 2470 ELSE H=1
5550 CONTROL S,2 ; 7@ ON TIMER# 3,30000 GOSUB 5890 @ DISP W0$ @ A7=1 @ GOTO 2470

5560 GOSUB 5960 @ IF H THEN 2470 ELSE OFF TIMER# 3 @ H=1
5570 DISP "Waiting for ring" @ DISP "Hit HANGUP to terminate Auto-Ans"
5580 K$="!" @ K0=0 @ ON KBD GOSUB 5690
5590 STATUS S,3 ; A@ IF K0 THEN 2470
5600 IF NOT BIT(A,3) THEN 5590
5610 WAIT 500 @ STATUS S,3 ; A
5620 IF NOT BIT(A,3) THEN DISP "Invalid ring!" @ WAIT 200 @ GOTO 5570
5630 ON TIMER# 3,17000 GOTO 5660 @ GOSUB 5710
5640 STATUS S,3 ; A
```

```
5650 IF A<>1 THEN 5640 ELSE OFF TIMER# 3 @ BEEP 100,75 @ BEEP 50,75 @ GOTO 2470
5660 OFF TIMER# 3 @ GOSUB 5900 @ GOTO 5570
5670 GOSUB 5960 @ IF H THEN 2470 ELSE H=1
5680 ON TIMER# 3,17000 GOSUB 5890 @ GOSUB 5710 @ DISP WO$ @ GOTO 2470
5690 K$=KBD$ @ IF K$=CHR$(131) OR K$=CHR$(139) THEN K$=K$&K$ @ KO=1
5700 ON KBD GOSUB 5690 @ RETURN
5710 HALT S @ CONTROL S,2 ; 1@ CONTROL S,2 ; 5@ RETURN
5720 GOSUB 5960 @ IF H THEN 2470 ELSE H=1
5730 DISP "Selftest started"
5740 OFF TIMER# 3 @ CONTROL S,2 ; 3@ WAIT 2000 @ STATUS S,3 ; A@ IF A=1 THEN 576
0
5750 DISP "Modem failed selftest" @ GOTO 5820
5760 STATUS S,9 ; A@ CONTROL S,9 ; A+4@ B1$,S$="Modem selftest" @ TRANSFER S$ TO
 S INTR
5770 STATUS S$,3 ; A@ IF A THEN 5770
5780 I$="" @ TRANSFER S TO I$ INTR ; COUNT LEN(B1$)
5790 WAIT 1000 @ HALT S @ IF B1$=I$ THEN DISP "Passed selftest" @ I$="" @ GOTO 5
820
5800 DISP "Selftest error" @ DISP "Sent:";B1$ @ DISP "Received:";I$ @ I$="" @ GO
TO 5820
5810 GOSUB 5290
5820 IF C4=1 THEN 5840
5830 IF H=0 THEN DISP "Modem off" @ GOTO 2470
5840 OFF TIMER# 3 @ M8=1 @ ON EOT S GOTO 5850 @ HALT S
5850 OFF EOT S @ ENABLE INTR S;0 @ CONTROL S,5 ; 0@ GOSUB 5860 @ GOTO 2470
5860 DISP "Hanging up" @ CONTROL S,2 ; 6@ WAIT 500 @ CONTROL S,2 ; 4@ H,L5,C4=0
5870 WAIT 1700 @ DISP "Modem off" @ RETURN
5880 K9=2 @ GOSUB 5300 @ GOTO 2470
5890 OFF TIMER# 3 @ STATUS S,3 ; A@ IF A MOD 2=1 THEN RETURN
5900 DISP "Time-out. No connection made." @ GOSUB 5860 @ RETURN
5910 B1$=P$ @ GOSUB 5930 @ IF P$="" THEN P$=B1$
5920 GOTO 2470
5930 ON ERROR GOTO 5940
5940 DISP "Phone number/name";@ INPUT P$@ RETURN
5950 GOSUB 5960 @ IF H THEN 2470 ELSE OFF TIMER# 3 @ L5=1 @ GOTO 5400
5960 IF H THEN DISP "Modem busy" @ BEEP @ RETURN
5970 GOSUB 5290 @ RETURN
5980 ! LOGON
5990 IF L>LEN(L$) OR L<=0 OR LO=0 THEN RETURN
6000 STATUS S$,0 ; RO,R1@ IF RO#R1+1 THEN RETURN
6010 IF A2>=D THEN RETURN
6020 ON POS("(PWE ",UPC$(L$[L,L]))+1 GOTO 6030,6050,6090,6120,6180,6185
6030 DISP "Invalid char in logon: ";L$[L,L] @ L=0 @ LO=0 @ RETURN
6040 ! "("
6050 L8=L-1+POS(L$[L,LEN(L$)],")") @ IF L8=L-1 THEN L8=LEN(L$)
6060 K$=K$&L$[L+1,L8-1] @ IF L$[L8,L8]=")" THEN K$=K$&L9$
6070 ON ERROR GOSUB 1620 @ K=LEN(K$) @ GOSUB 2530 @ IF L<>0 THEN L=L8+1 @ RETURN
 ELSE RETURN
6080 ! "P"
6090 IF L<>0 THEN DISP "Logon paused. <RUN> to proceed" @ L=L+1 @ LO=0
6100 RETURN
6110 ! "W"
```

```
6120 IF POS("0123456789",L$[L+1,L+1])=0 THEN ON SGN(L)+1 GOTO 6100,6030
6130 IF L=0 THEN RETURN ELSE L8=VAL(L$[L+1,L+1]) @ L=L+2
6140 IF L8=0 THEN L8=10
6150 IF L=0 THEN RETURN ELSE ON TIMER# 3,L8*1000 GOSUB 6160 @ L0=0 @ RETURN
6160 OFF TIMER# 3 @ L0=1 @ RETURN
6170 ! "E"
6180 IF L=0 THEN RETURN ELSE L=L+1 @ K$=K$&L9$ @ L0=1 @ K=LEN(K$) @ GOSUB 2490 @
 RETURN
6185 IF L=0 THEN RETURN ELSE L=L+1 @ RETURN ! " "
6190 GOTO 7400
6200 K4=LEN(B1$)
6210 K4=15*K1-K4
6220 FOR J=1 TO K4
6230 B1$=B1$&" "
6240 NEXT J
6245 K1=K1+1
6250 RETURN
7000 K1=1
7010 READ# 1,2 ; Y,D2,T2,D3,T3
7020 B1$=VAL$(Y)
7030 GOSUB 6200
7040 B1$=B1$&VAL$(D2)&"."
7050 GOSUB 6200
7060 B1$=B1$&VAL$(T2)
7070 GOSUB 6200
7080 B1$=B1$&VAL$(D3)&"."
7090 GOSUB 6200
7100 B1$=B1$&VAL$(T3)&L9$
7110 GOSUB 3430
7115 J7=1
7116 IF J7>Y THEN 3680
7130 READ# 1,J7+2 ; T,D,F1(),D4()
7140 K1=1
7150 B1$=VAL$(D)&"."
7160 GOSUB 6200
7170 B1$=B1$&VAL$(T)&L9$
7180 GOSUB 3430
7185 K3=1
7190 FOR K2=1 TO 16
7200 K1=1
7210 B1$=""
7220 FOR K=1 TO 2
7230 B1$=B1$&VAL$(D4(K3))&"."
7240 GOSUB 6200
7250 B1$=B1$&VAL$(F1(K3))
7260 GOSUB 6200
7270 K3=K3+1
7280 NEXT K
7290 B1$=B1$&L9$
7300 GOSUB 3430
7310 NEXT K2
7320 J7=J7+1
7321 GOTO 7116
7400 END
```

U.S. Department of the Interior
Bureau of Mines—Prod. and Distr.
Cochrans Mill Road
P.O. Box 18070
Pittsburgh, Pa. 15236

□ Do not wish to receive this material, please remove from your mailing list.

□ Address change. Please correct as indicated.